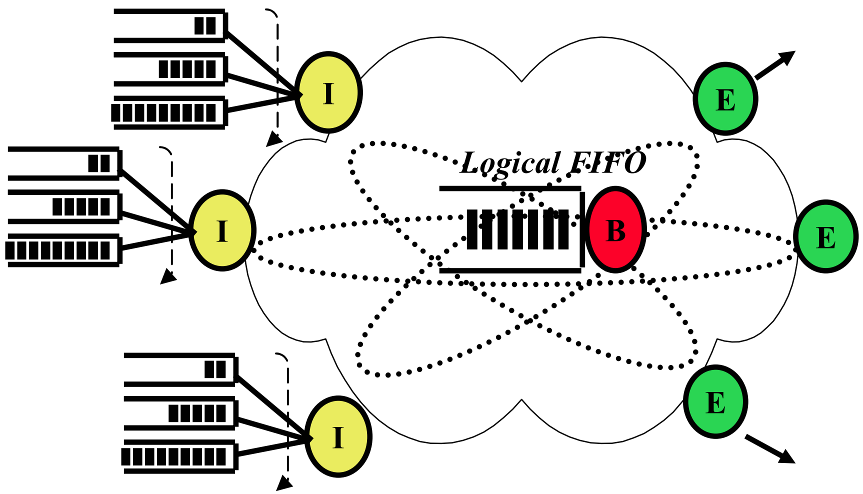
Overlay QoS using Closed-Loop Control: Expected Minimum Rate Service



David Harrison, Yong Xia, Arvind Venkatesh, Shiv Kalyanaraman,

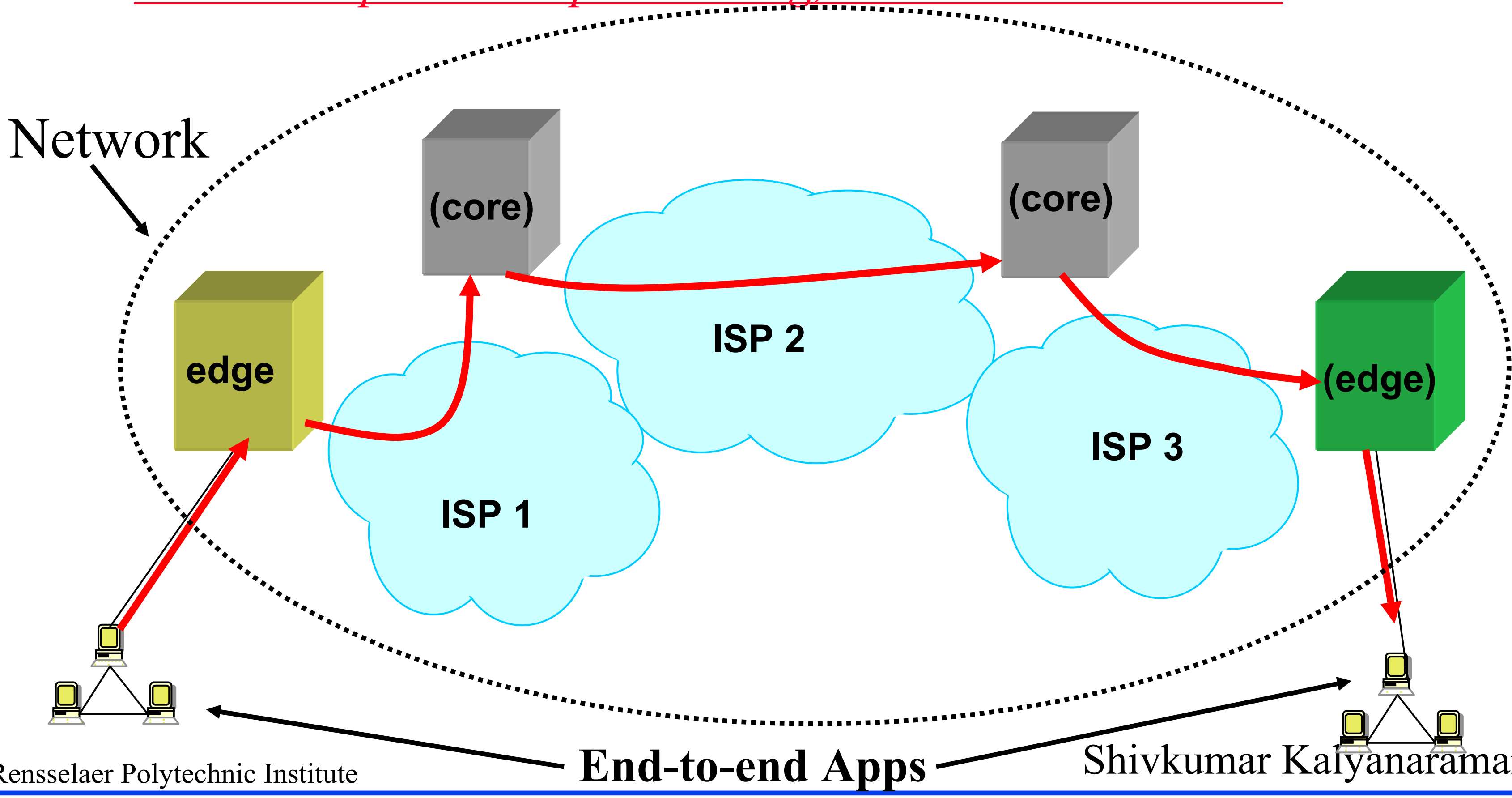
Rensselaer Polytechnic Institute

shivkuma@ecse.rpi.edu

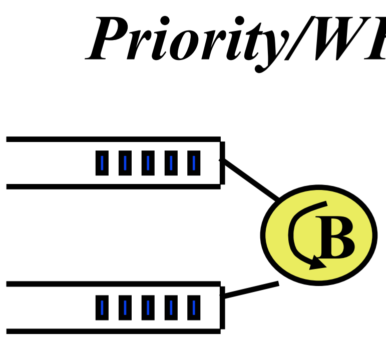
http://www.ecse.rpi.edu/Homepages/shivkuma

Big Picture: Overlay Network Services

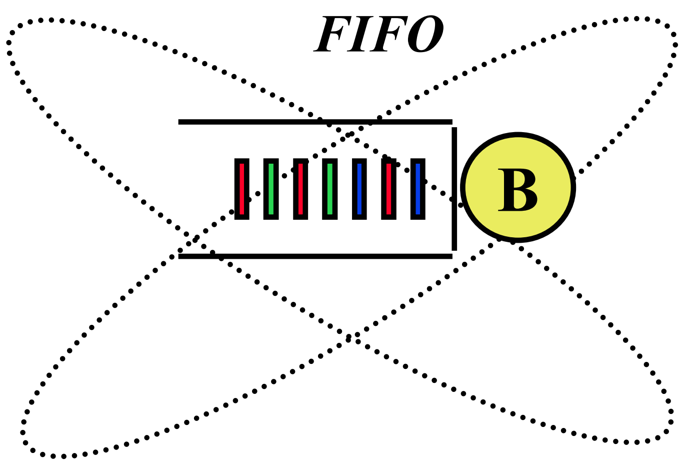
- *Lightweight* network svcs (eg: QoS, multi-paths) can dramatically enhance application-perceived performance
  - Overlay => such services in a *multi-provider* environment, or
  - *Dramatically reduced complexity* of network services in a *single provider*
  - *Distributed parameter provisioning, no admission control…*



What is Closed-loop QoS? (Qualitatively)



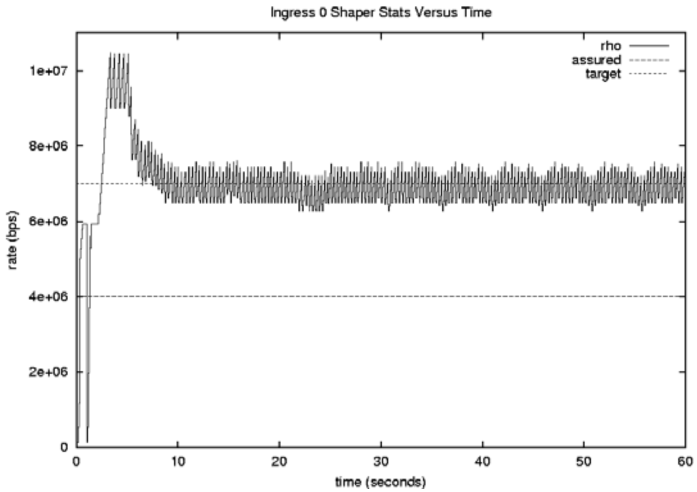
$\Rightarrow$



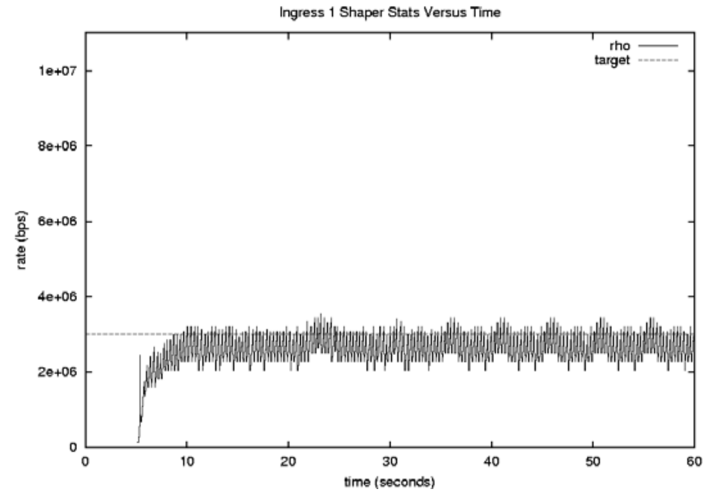
- **<u>Scheduler:</u>** differentiates service on a ***<u>packet-by-packet</u>*** basis

- **<u>Loops:</u>** differentiate service on an ***<u>RTT-by-RTT</u>*** basis using ***edge-based policy configuration***.
  - Differentiation/Isolation meaningful <u>in steady state only</u>…

Expected Min Rate (EMR) Service: Sample Steady State Behavior



Flow 1 with 4 Mbps assured + 3 Mbps best effort



Flow 2 with 3 Mbps best effort

QoS spectrum

Best Effort ... **Leased Line**

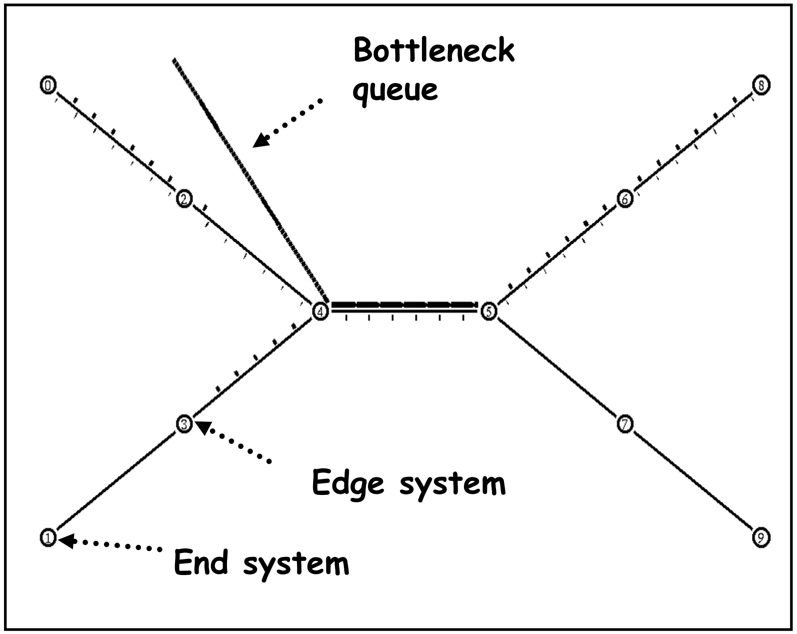
OverQoS, Overlay QoS w/ Closed Loop Control

DPS/CSFQ, Diff-Serv

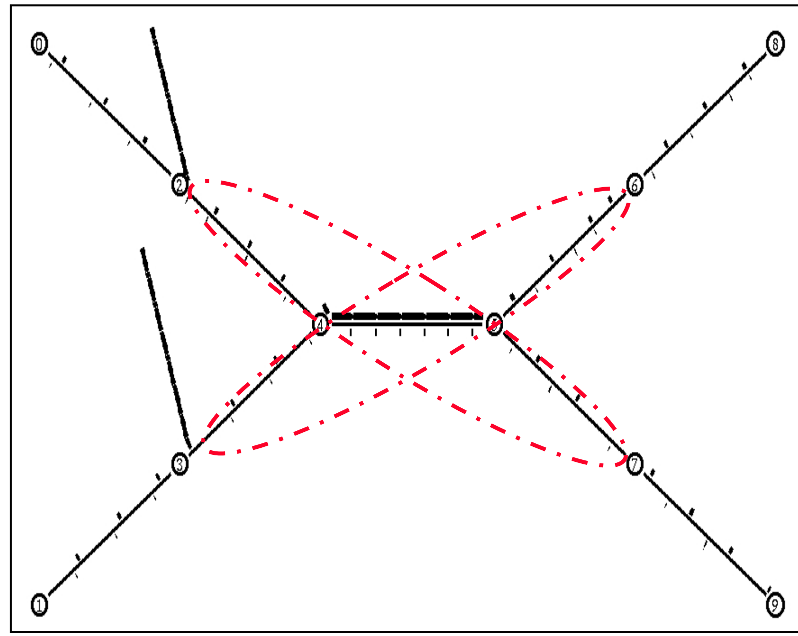
Int-Serv

Architectural Advantages of Closed Loops

- Traffic management consolidated at edges (placement of functions in line with E2E principle)

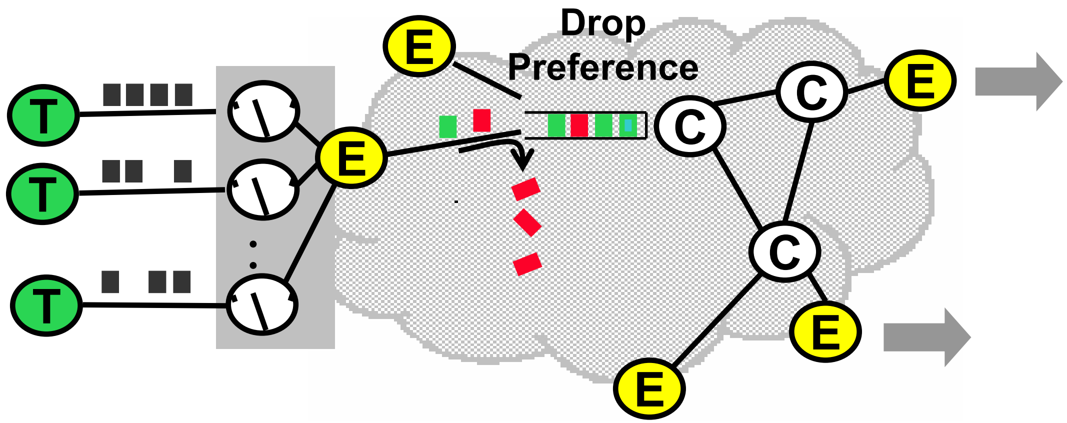


⇒

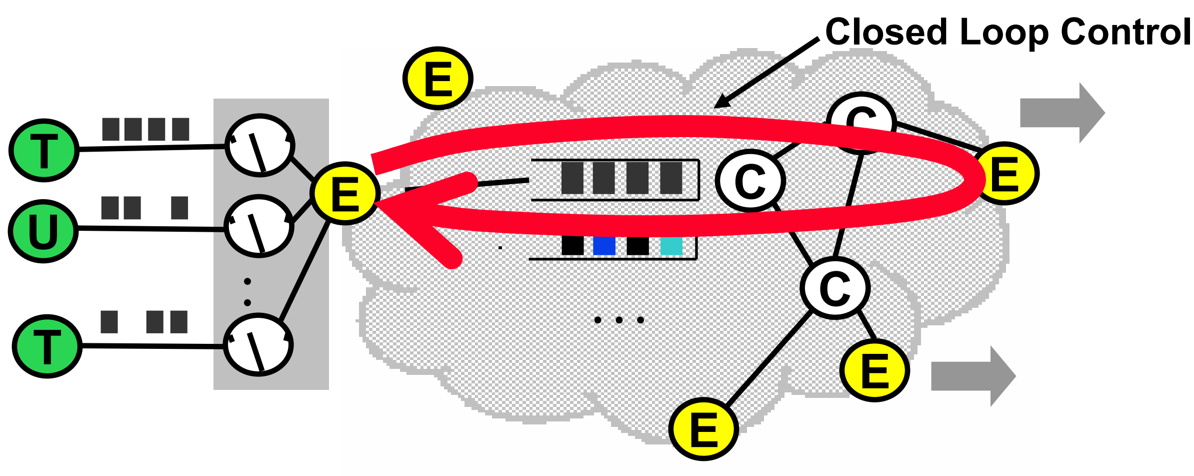


- **Architectural Potential:**
  - Edge-based (distributed) QoS services,
  - Edge plays in application-level QoS

Diff-Serv vs Closed-loop QoS

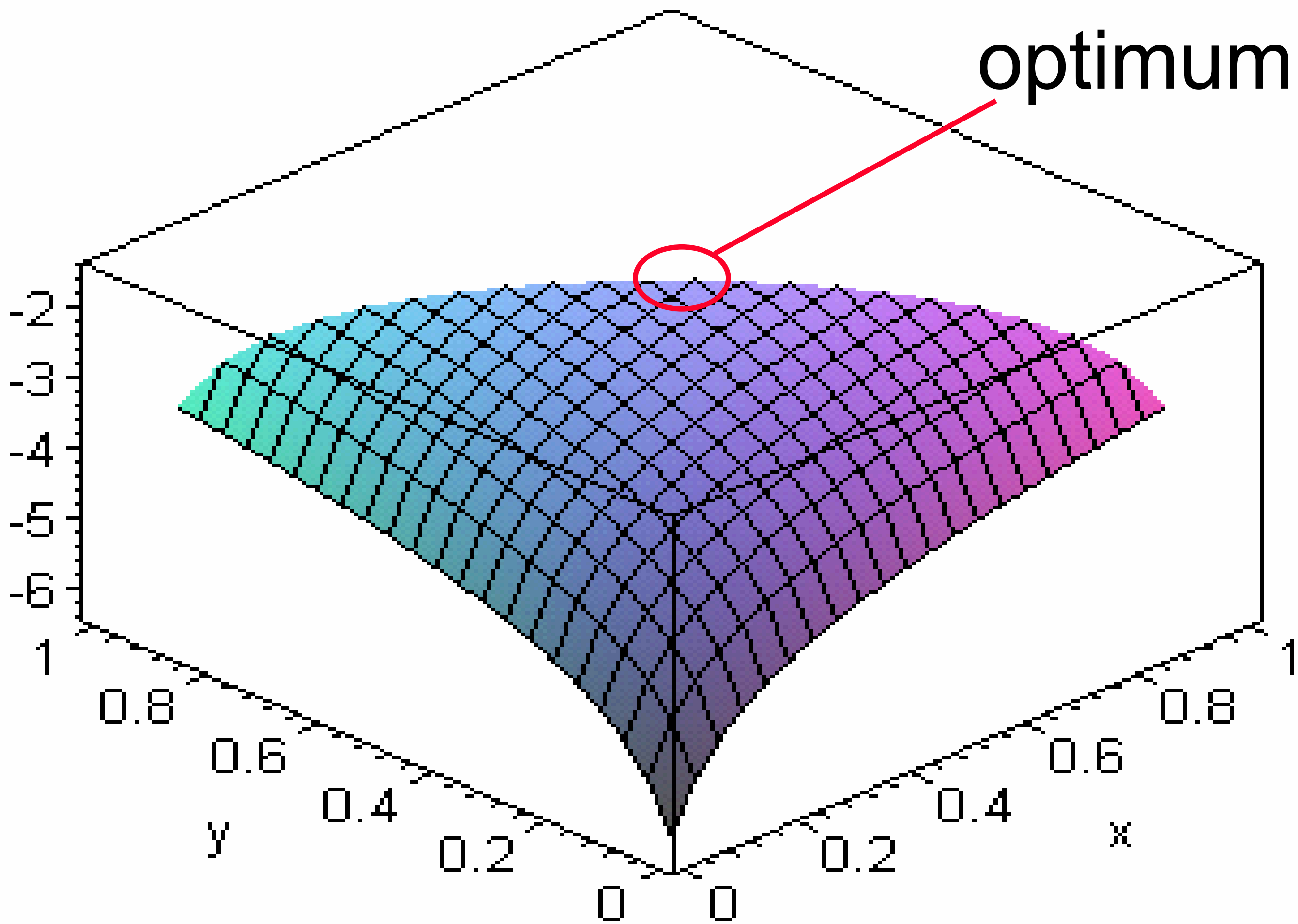


vs



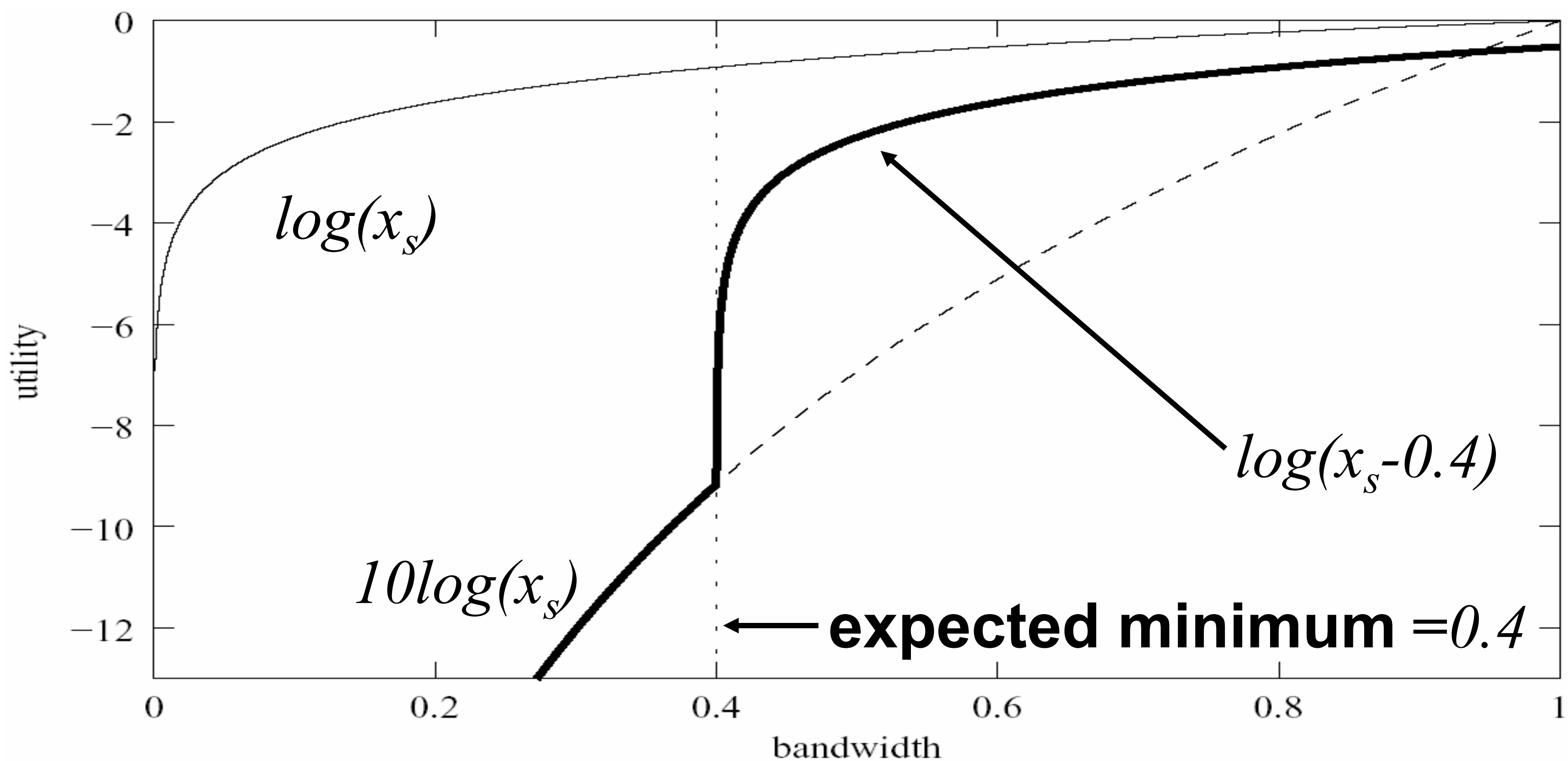
Kelly's Framework: Illustration

Maximize $U(x)+U(y)=\log(x)+\log(y)$



Issue: QoS => Non-concave User Utility Functions

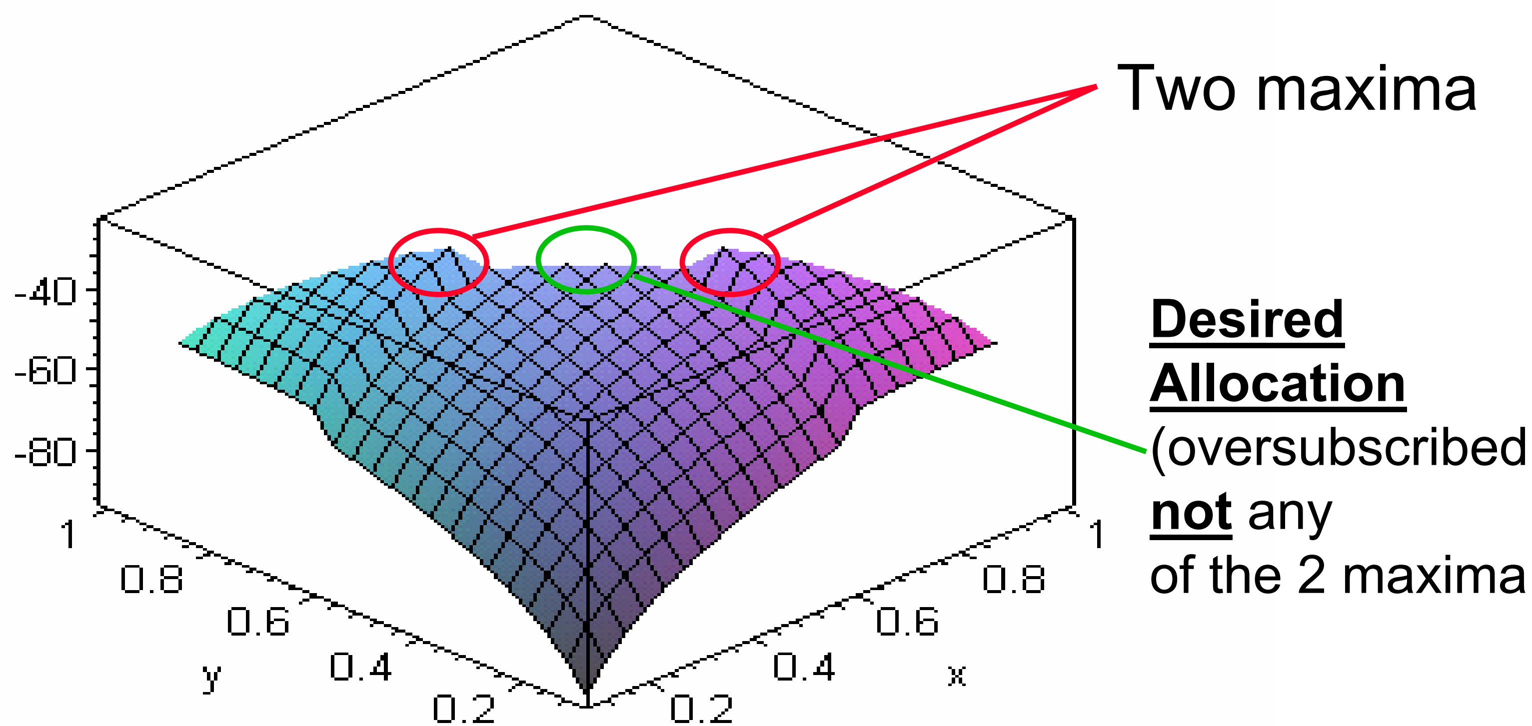
- A user with a *minimum rate* QoS expectation (*gracefully degrading into a weighted service*) can be modeled with a *non-concave* utility function.
- But this kind of U-function cannot be plugged into Kelly's non-linear optimization formulation directly!



Luckily, the Sum of Non-Concave U-fns is not what we want to Optimize!

- U(z) = log(*z-0.6*) if z > 0.6 ***(expected minimum rate)***

  10logz if z <= 0.6 ***(graceful degradation to weighted svc)***



• Can use strictly concave functions and define multiple optimization problems for the same QoS problem &

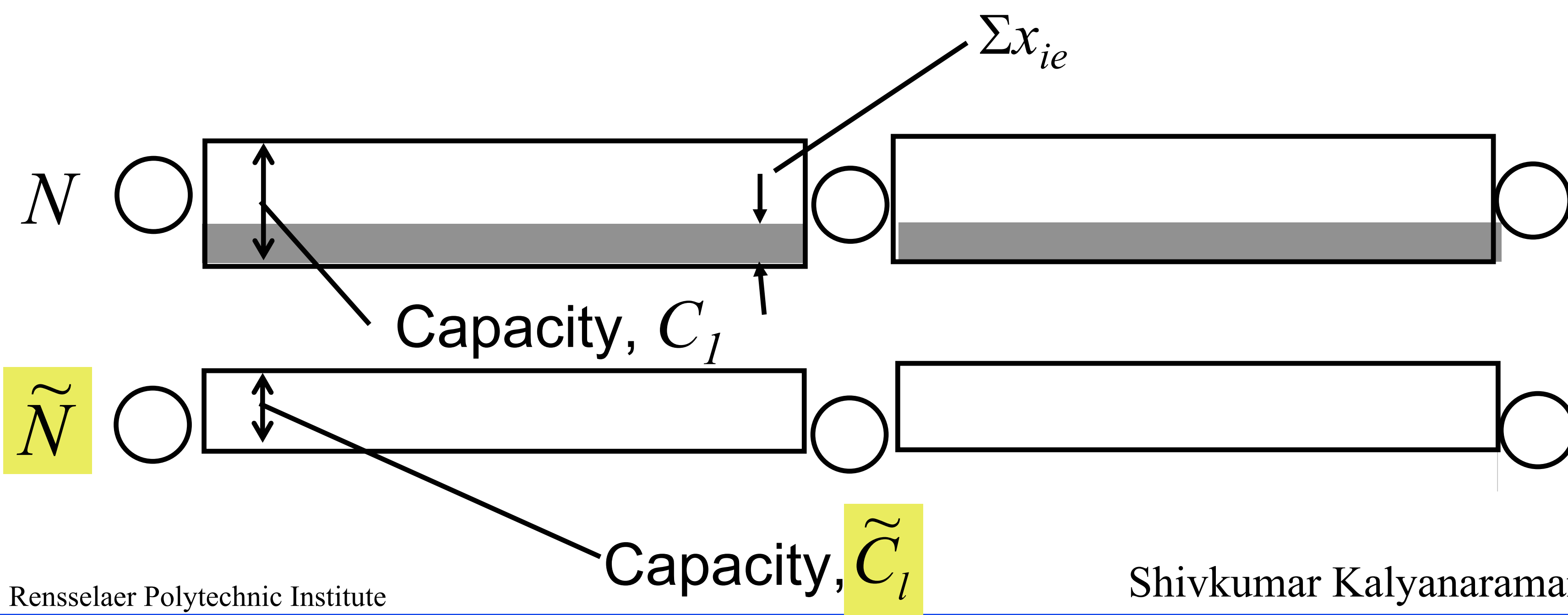
• Dynamically choose a different optimization problem when oversubscribed

No Over-Subscription Case: ***Auxiliary*** Problem

- Let $x_{ip} = x_i - x_{ie}$ i.e. flow i: two ***virtual sub-flows on same path***
- Think of a modified network $\tilde{N}$ with modified link capacities

$$\tilde{C}_l = C_l - \sum x_{ie}$$

Provide ***proportional fairness*** on ***residual*** network capacity



Handling both *under-* and *over*-subscription…

- For a_i, x_i: (primary problem)

$$\begin{aligned} &\text{maximize} && \sum_{i \in I} a_i \ln x_i \\ &\text{subject to} && \sum_{i \in I_l} x_i \le c_l, \ \forall l \in L \\ &&& x_i > 0, \ \forall i \in I \end{aligned}$$

Effective when:

$a_i = A_i$

$$q_l \le Q_l, \quad \forall l$$

- For a_{ip}, x_{ip}: (auxiliary problem)

$$\begin{aligned} &\text{maximize} && \sum_{i \in I} a_{ip} \ln x_{ip} \\ &\text{subject to} && \sum_{i \in I_l} x_{ip} \le c_l - \sum_{i \in I_l} x_{ie}, \ \forall l \in L \\ &&& x_{ip} > 0, \ \forall i \in I \end{aligned}$$

Effective when:

$$\sum_{i \in I_l} x_{ie} < c_l, \quad \forall l$$

$$q_l \le Q_l, \quad \forall l$$

$a_i < A_i$

If under-subscribed, solve the aux-problem; and the primary problem is automatically solved (note: a_{ip} = constant)

Accumulation-Based Congestion Control

Key idea: develop a notion of "accumulation" (a_i or a_{ip}) as a steering parameter for QoS

Why accumulation? Why not just use weighted AIMD?

- Loss-based CC fails to provide large range of QoS capabilities
- Couples transient dynamics of CC with equilibrium specification
- Interacts with TCP reliability mechanisms (eg: timeout)

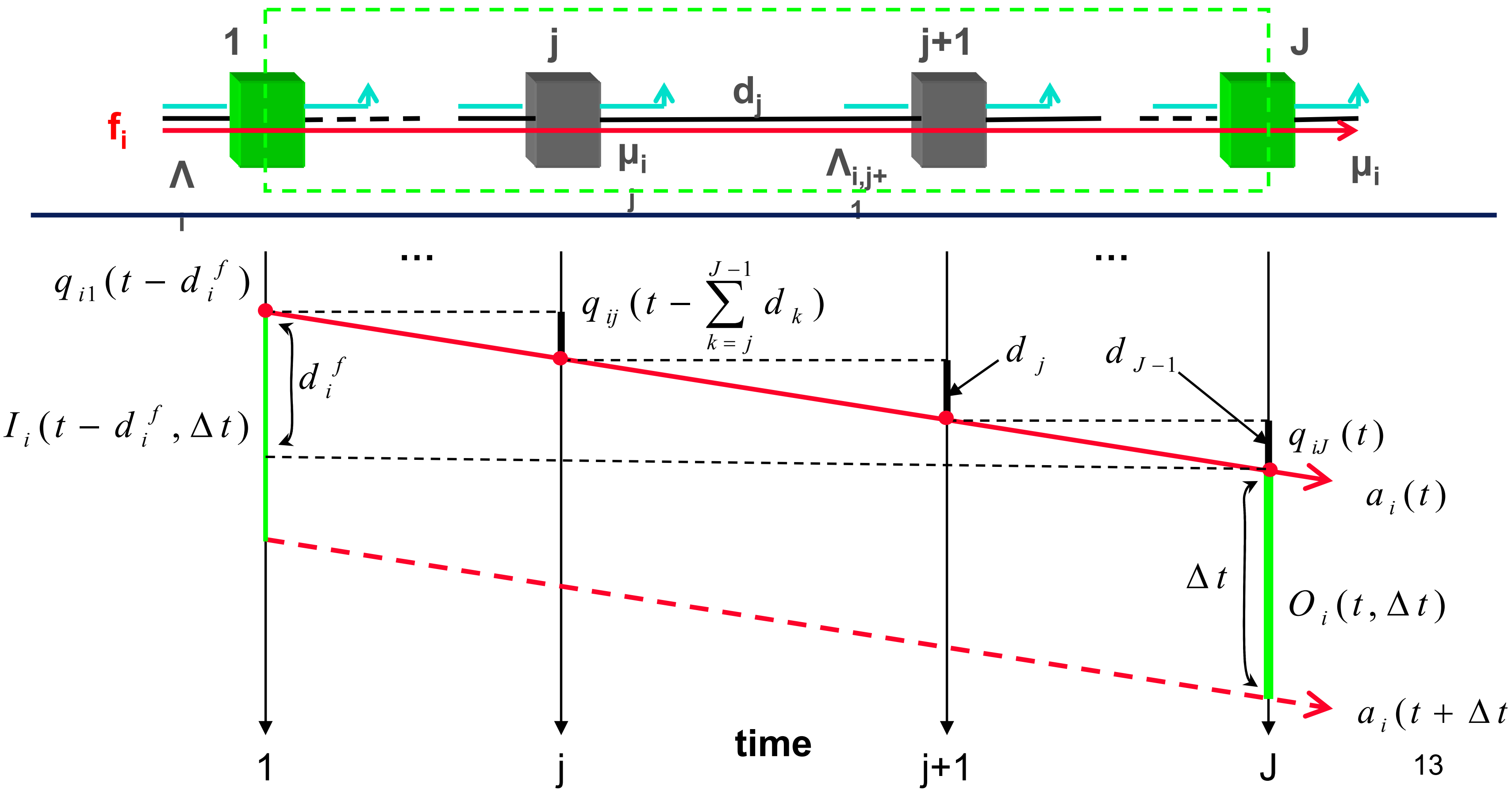
Why not ECN or AQM schemes?

- Want to keep AQM support as optional, not mandatory

Why not use just Vegas?

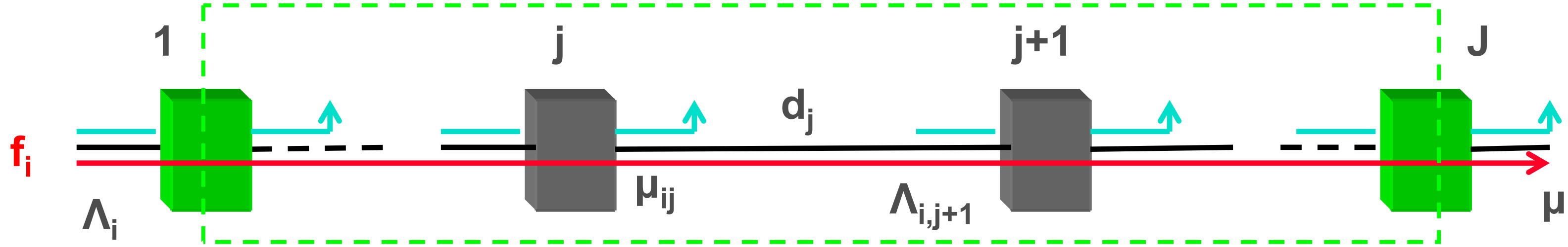
- Accumulation is an abstract dynamical concept.
- Vegas and Monaco attempt to provide estimators for accumulation.
- Vegas' accumulation estimator is not robust

Accumulation: Definition & Physical Meaning



$$a_i(t) = \sum_{j=1}^{J} q_{ij}(t - \sum_{k=j}^{J-1} d_k)$$

Accumulation-based Control Policy



- control objective : keep $a_i(t) = a_i^* > 0$
  - if goal $a_i(t) = 0$, no way to probe increase of available b/w;
- control algorithm :

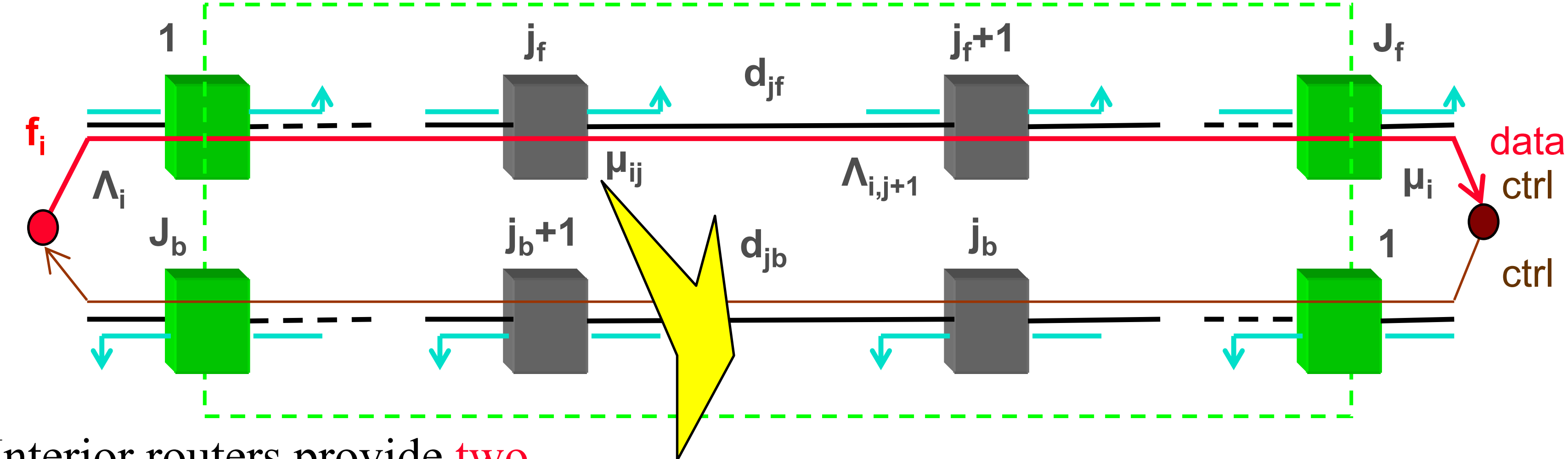
$$\begin{aligned}
&if \quad a_i(t) < a_i^* \quad then \quad \lambda_i \uparrow \\
&if \quad a_i(t) > a_i^* \quad then \quad \lambda_i \downarrow \\
&recall \quad : \Delta a_i(t, \Delta t) = [\overline{\lambda}_i(t - d_i^f, \Delta t) - \overline{\mu}_i(t, \Delta t)] \times \Delta t
\end{aligned}$$

- Example control algorithm :

$$\dot{w}_i(t) = -k \cdot f(a_i(t) - a_i^*)$$

$$where \quad f \uparrow, \quad only \quad f(0) = 0, \quad k > 0$$

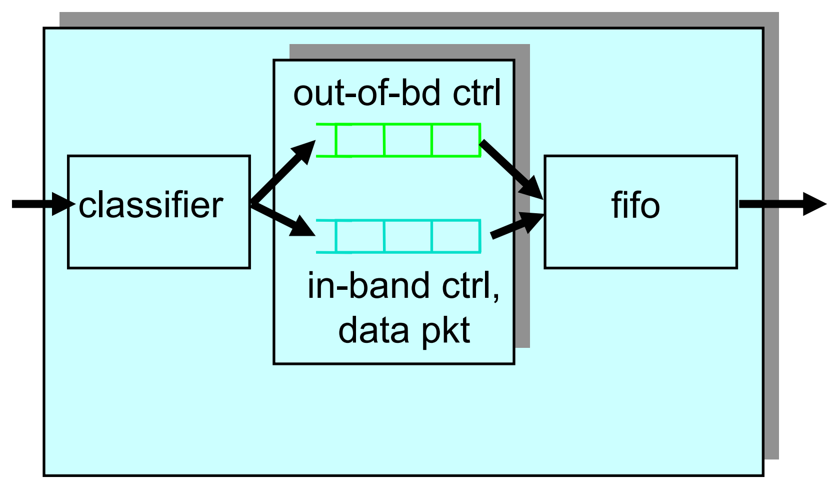
Monaco Accumulation Estimator



Interior routers provide two priority fifo queues :

1) high priority queue for **out-of-band** control packet

2) low priority queue for **in-band control packet and data packet**



Can be done w/ IP precedence on existing routers in Internet!!

ACC: Monaco vs Vegas (estimation robustness)

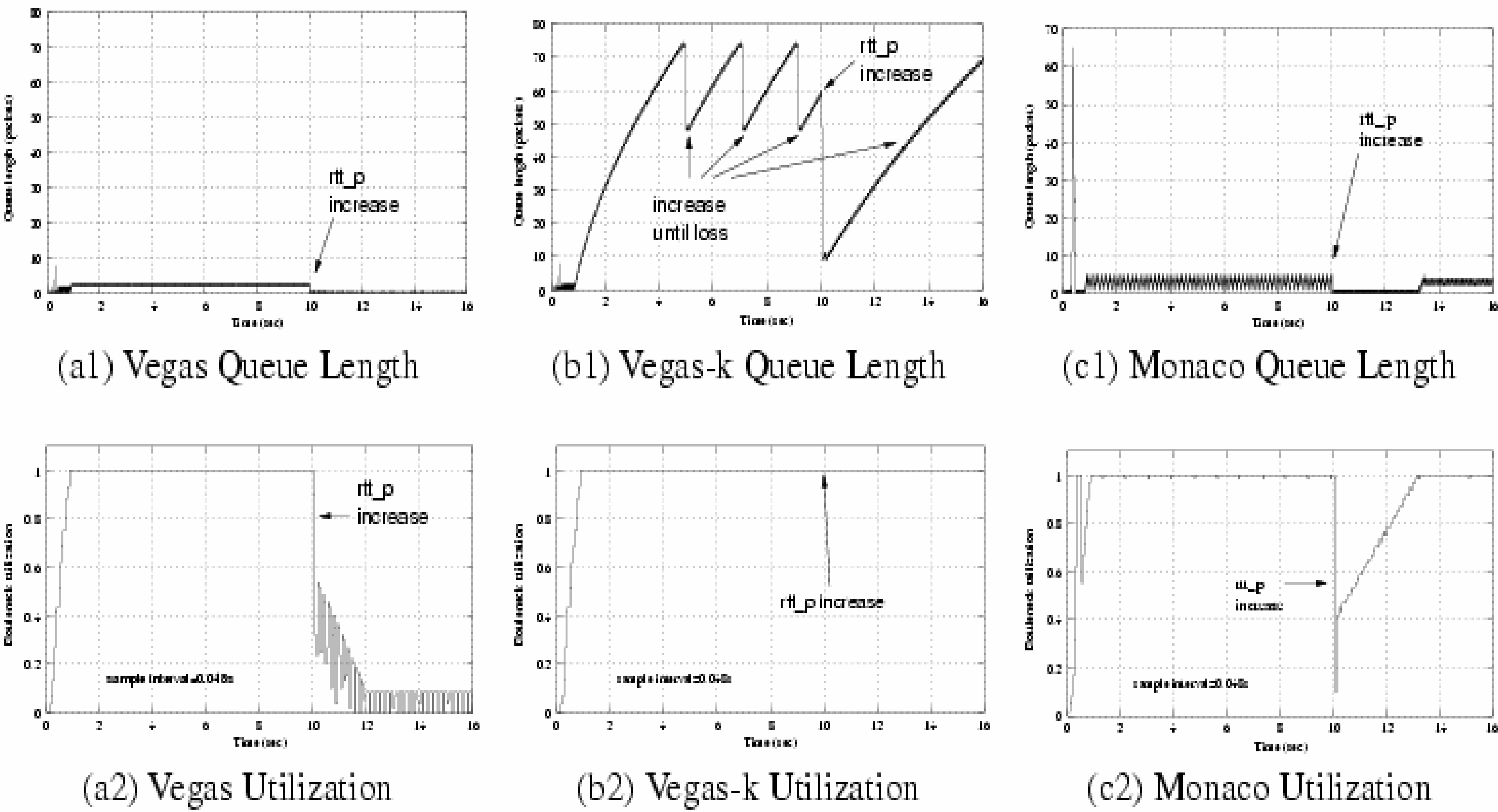


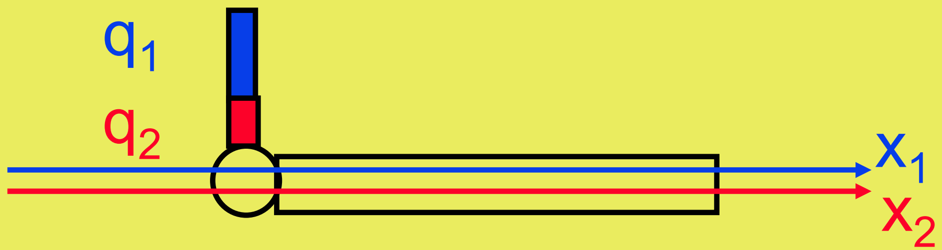
Fig. 4. Comparison between Vegas, Vegas-k and Monaco under rtt_p(or $basertt$) Estimation Error

Key Notion: "Accumulation"

- Accumulation-based congestion control (ACC) is a nonlinear optimization, where user i maximizes: $U_i(x_i) = a_i \ln x_i$
  - Accumulation (a_i) is the ***weight*** (w_i) of the weighted prop. fair allocation
- Accumulation is hence a "***steering***" parameter:
  - Equilibrium *accumulation* allocation => Equilibrium *rate* allocation!
  - Dynamics of CC scheme ***decoupled*** from equilibrium spec (unlike AIMD)
- Accumulation has a ***physical*** meaning: sum of buffered bits of the flow in the path
- Accumulation is related to the ***lagrange multiplier***, I.e., $\mathbf{a_i = \Sigma p_l}$

- For two flows i,k sharing the ***same*** path, $\mathbf{a_i / a_k = x_i / x_k}$
  - FIFO queues => arrival order decides departure order

  => buffer occupancy decides rate allocation



Over-subscription: Key Idea

- The virtual sub-flows x_i, x_{ie}, x_{ip} are on the same path (same real flow!):

$$\left.\begin{array}{l} \boxed{x_{ie} + x_{ip} = x_i, \quad \forall i} \\ \boxed{a_{ie} + a_{ip} = a_i, \quad \forall i} \end{array}\right\} \qquad \boxed{\frac{a_{ie}}{x_{ie}} = \frac{a_{ip}}{x_{ip}} = \frac{a_i}{x_i}, \quad \forall i} \quad \ldots \text{(I)}$$

- And,

$$\boxed{a_{ip} = const = a_i - a_{ie} = a_i\left(1 - \frac{x_{ie}}{x_i}\right), \quad \forall i} \quad \ldots \text{(II)}$$

- a_i, x_i are measurable, $x_{ie} = \tilde{x}_{ie}$ (contracted rate), if under-subscribed
- During over-subscription, $\sum_{i \in I_l} \tilde{x}_{ie} \geq c_l, \quad \exists l$
- Since a_{ip} = constant, eqn (II) implies that $\uparrow x_i$, $\uparrow a_i$ unboundedly
- But $a_i <= A_i$
  - The auxiliary problem ***drops out*** for some flows (eg: bronze flows) and
  - Their rate is determined by the primary problem

    (I.e. gracefully degraded to a weighted proportional fair allocation)

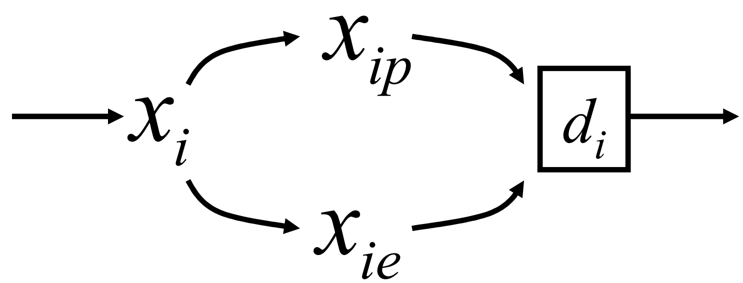
EMR Building Block

- **Accumulation**

$$a_i = x_i d_i$$

$$a_{ie} = x_{ie} d_i$$

$$a_{ip} = (x_i - x_{ie}) d_i$$



Accumulation limit

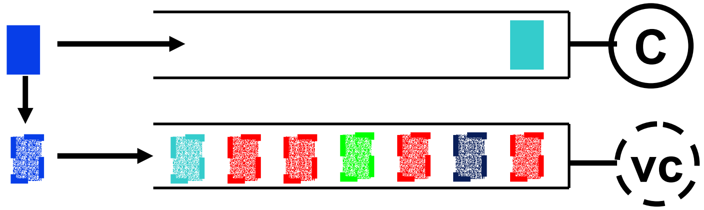
Target

- **Control Law**

$$\boxed{\Delta w_i = -\,\kappa \max(a_{ip} - a^{*}_{ip},\ a_i - A_i)}$$

Estimated accumulation
in virtual network

Virtual Accumulation (with AQM): Integration with UIUC Work (Srikant)



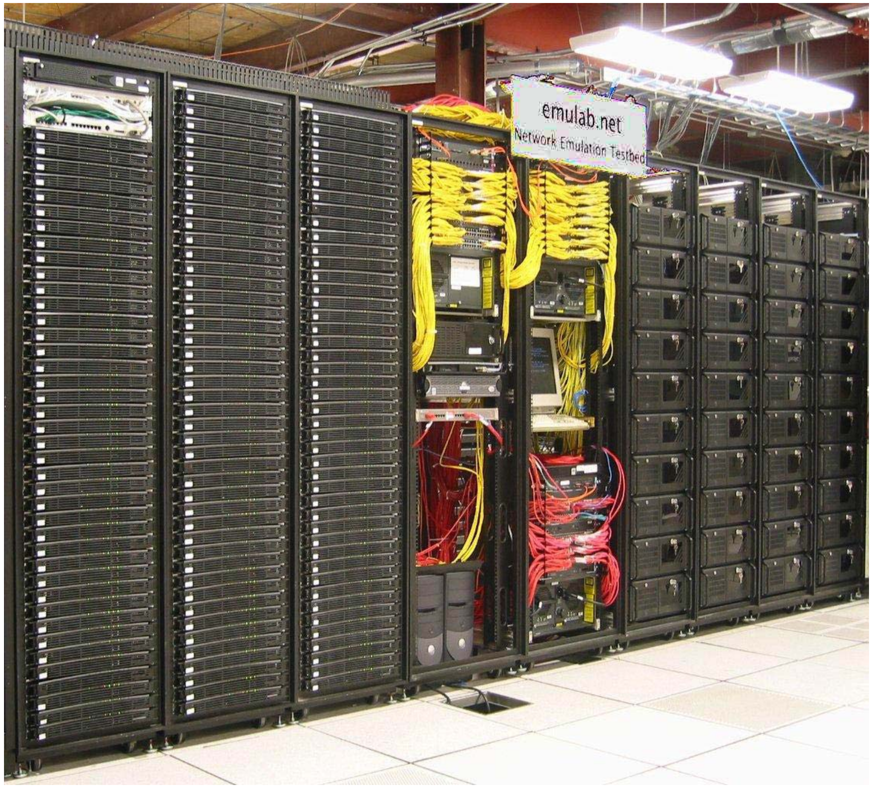
- Use virtual queueing delay, *vd*.

  vd=vq/vc. *(eg: by modifying AVQ)*
- Communicate *vd* in probe packets (add vds on path).
- *Accumulation = physical + virtual accumulation*

$$a_i = x_i(d_i + \sum vd_l)$$

- Both AQM and non-AQM nodes in same network.

Simulation/Implementation/Testing Platforms

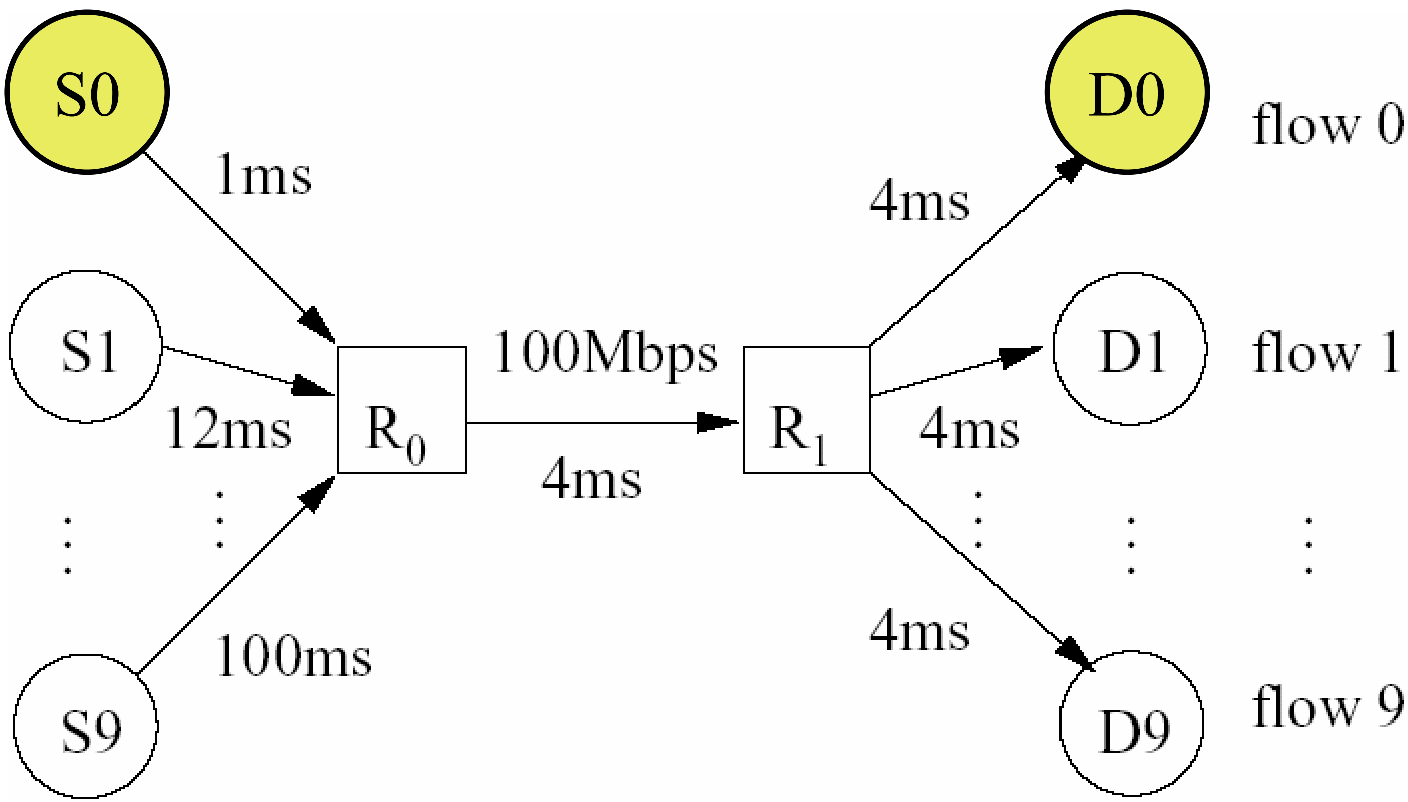


Utah's **Emulab** Testbed:
Experiments with
Linux/Zebra/Click
implementation

MIT's **Click Modular Router**
On Linux:
Forwarding Plane

Modular *Router*

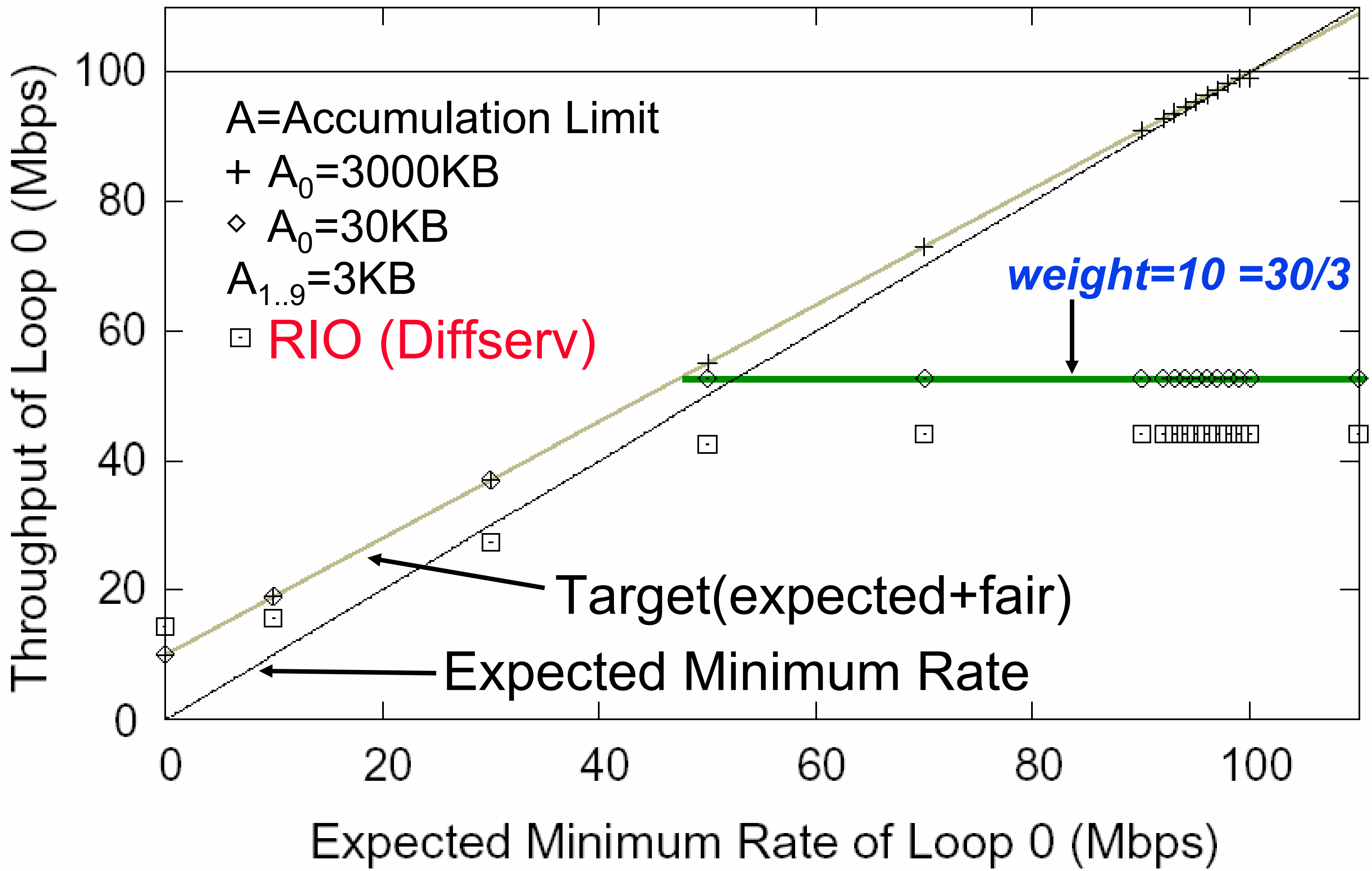
Single Bottleneck Topology



All links are 100Mbps.
S=Source. D=Destination. R=Router.

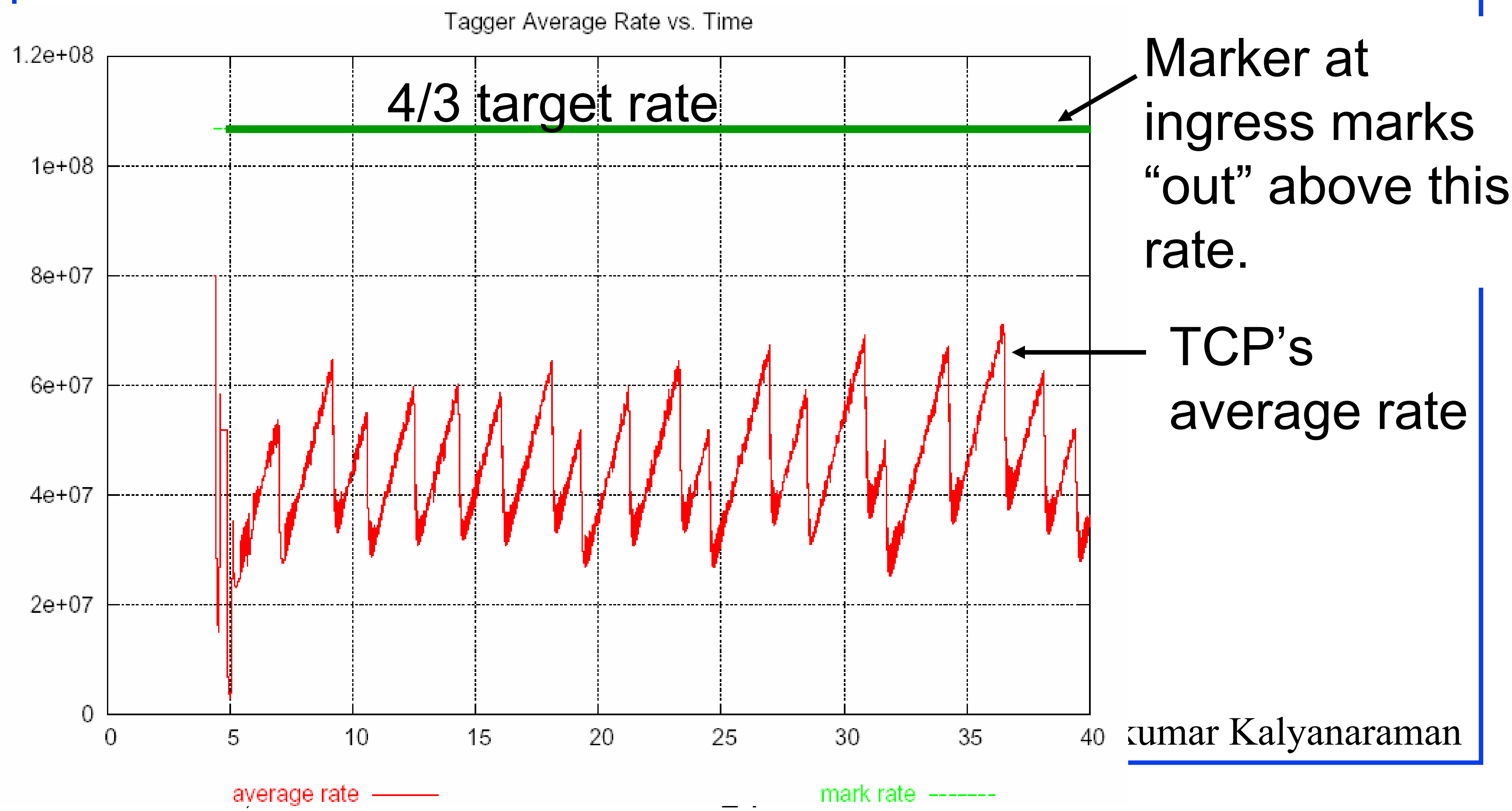
S0-D0 offered an expected minimum rate

Range of Expected Minimum Rates

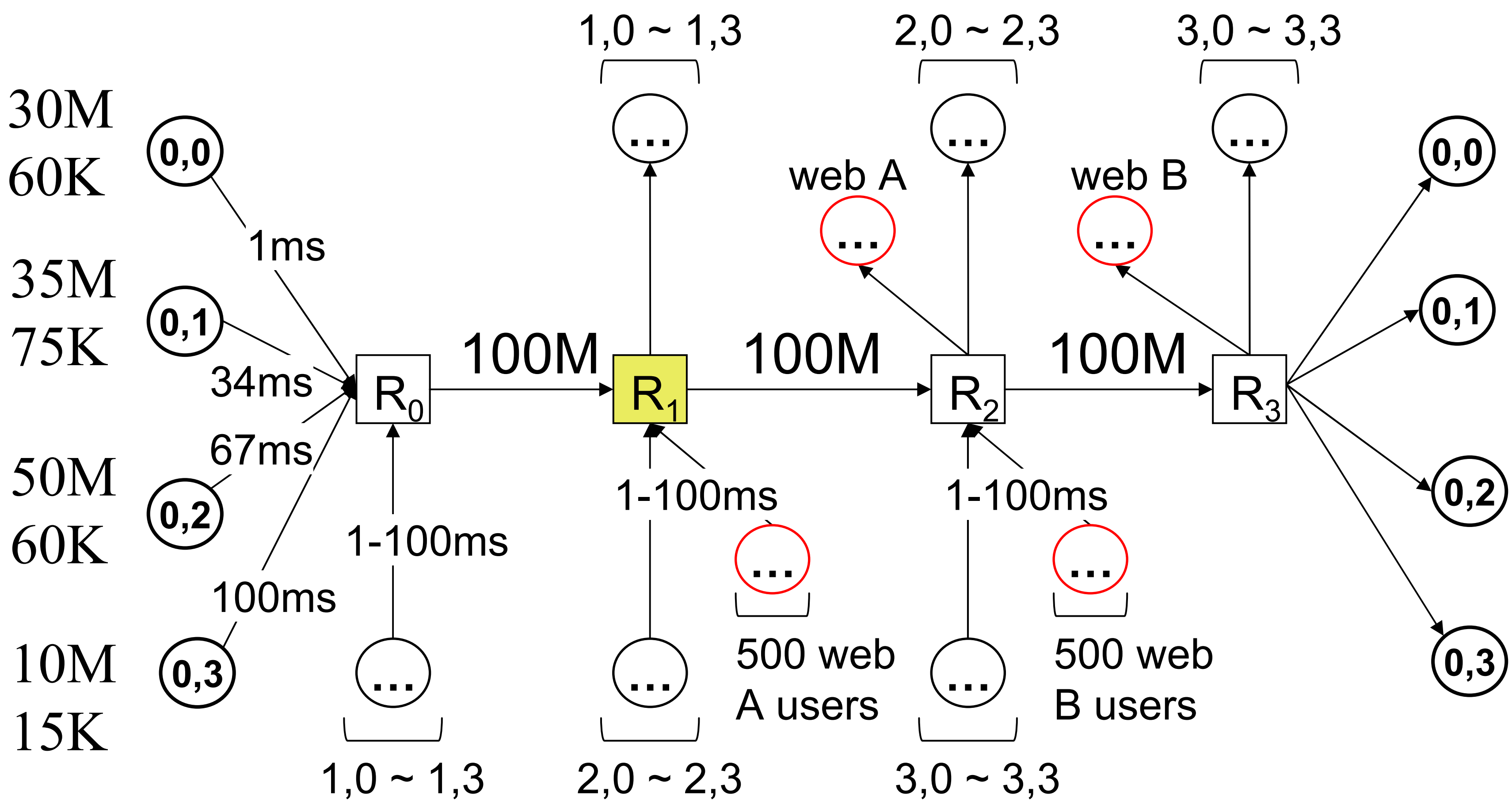


Compared to Diffserv AF (TCP+RIO)…

- Size of TCP oscillations increases with send rate.
- Achieving high assurances requires re-parameterizing bottleneck to permit large queues.



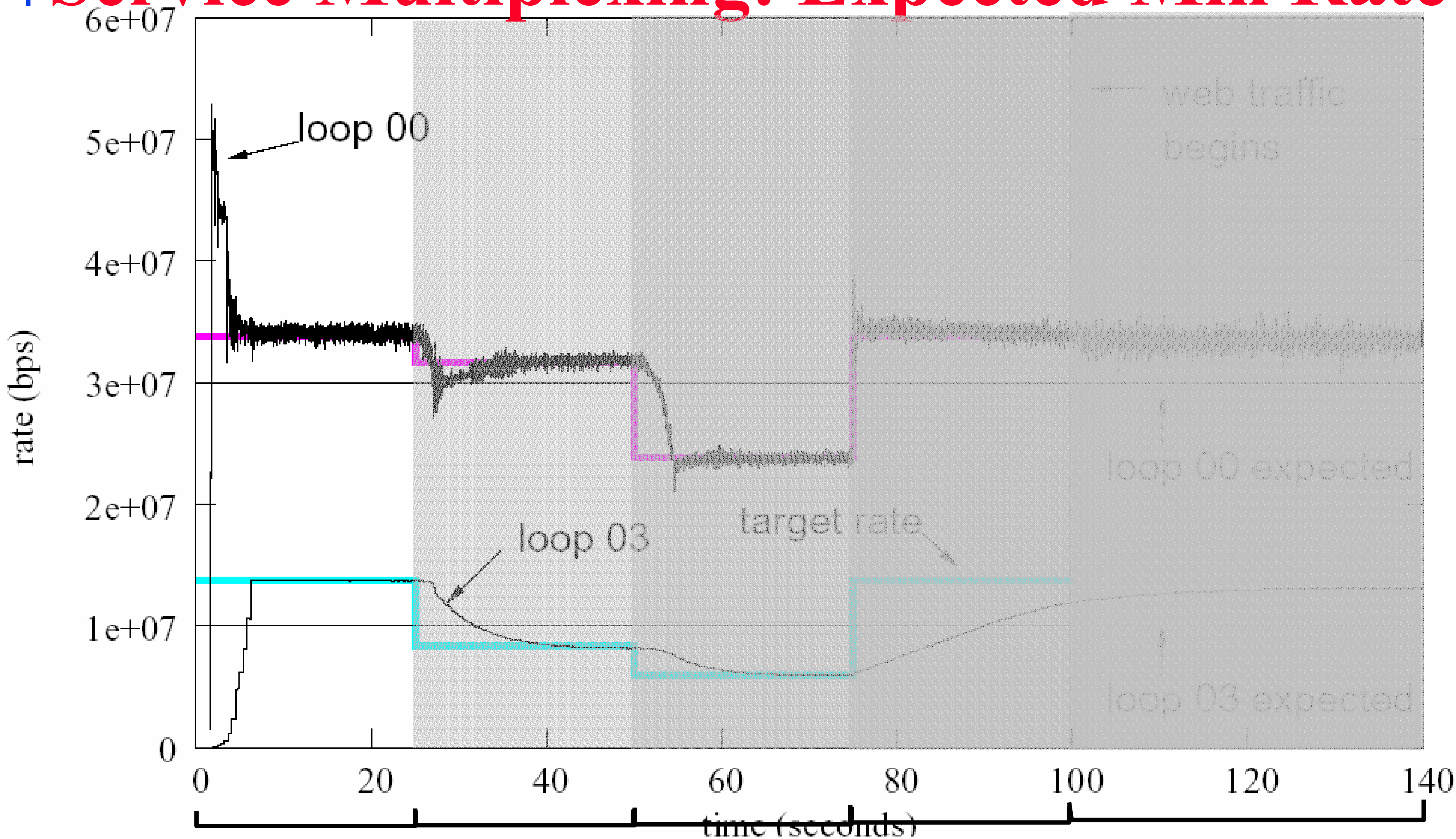
Service Multiplexing Topology



2,0 has weight 3

- Bandwidth for all unlabelled links are 1Gbps; Delay 1ms;
- **AQM+VD at router R1**, no AQM at other routers

Service Multiplexing: Expected Min Rate



No oversu
$<m_{00}=30,A$
$<m_{03}=10,A$

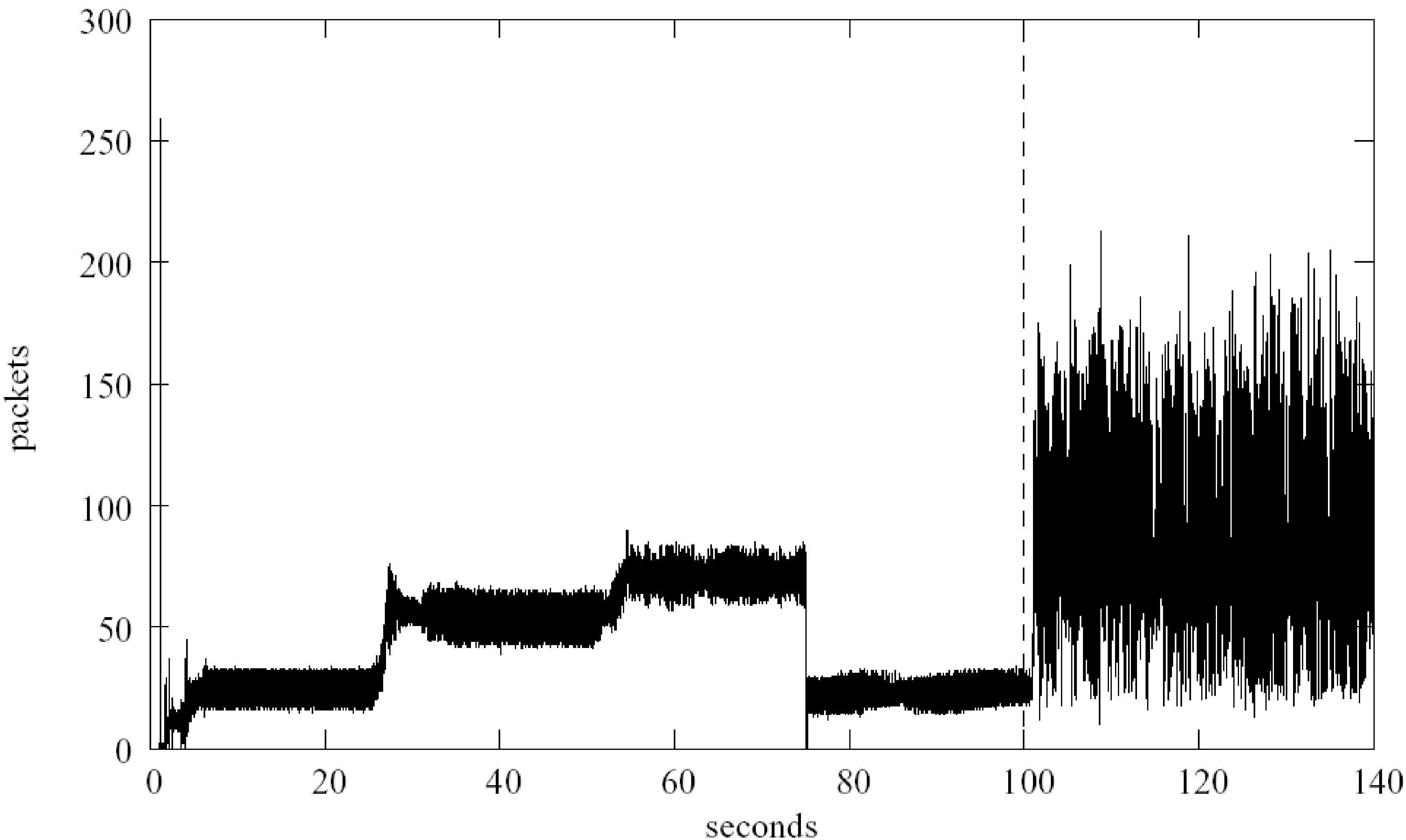
Moderate ove
$<m_{01}=35,A_{01}=$

Gross oversubscription
$<m_{02}=50,A_{02}=60KB>$

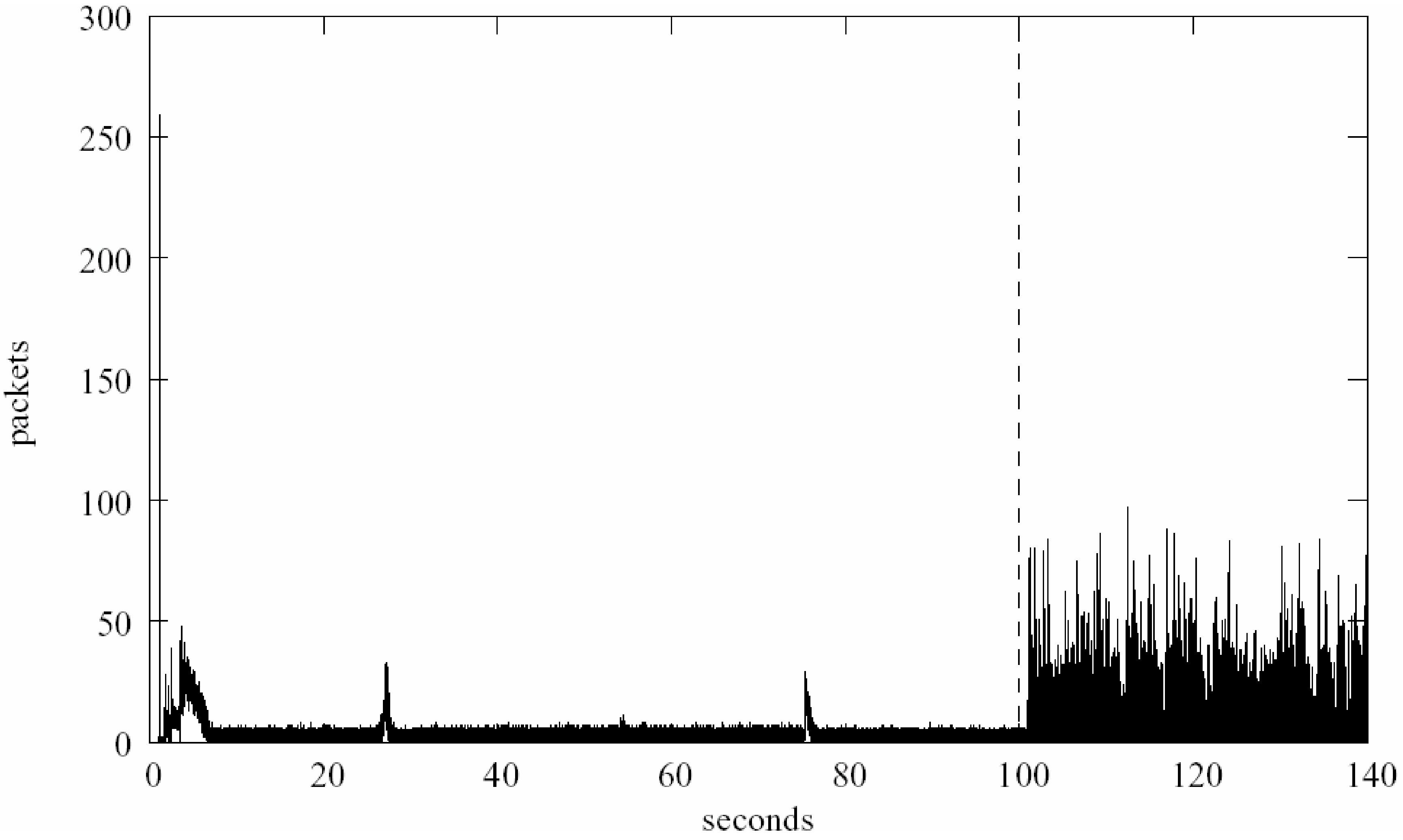
Loop 02,03 stop sending

Web

Non-AQM Router Queue Length



AQM+Virtual Accumulation Queue Length



Summary

<u>QoS</u> can be viewed as a <u>congestion control</u> problem

and therefore,

QoS can be posed in Kelly's optimization framework

<u>Challenges:</u>

1. What about the non-concavity of QoS utility functions?
2. Can we do away with admission control ?

<u>Ans:</u>

1. Define & Solve an Auxiliary Optimization Problem
2. Alternative Convex Constraints in Lagrange Domain can avoid need for admission control, allowing graceful service degradation

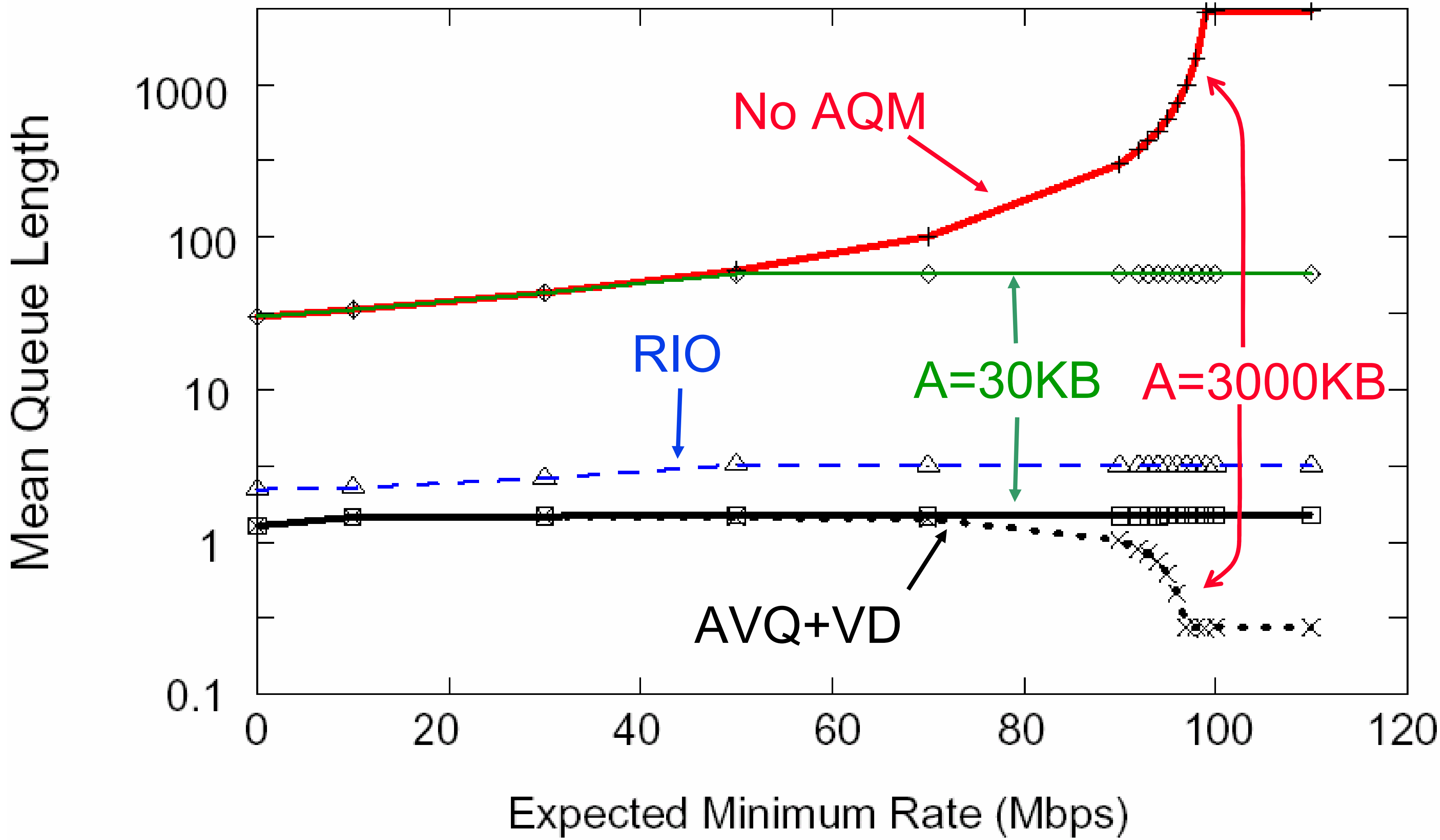
Future Work

- Distributed parameter setting guidelines w/o admission control
- Broader set of service semantics
- Deployment on PlanetLab
- Multi-ISP issues:
  - Data-plane: variable delay virtual links
  - Control-plane: accounting, SLA verification, minimal signaling architecture
- Overlay QoS in multi-hop wireless networks
- Applications: interactive/streaming video, VoIP over e2e

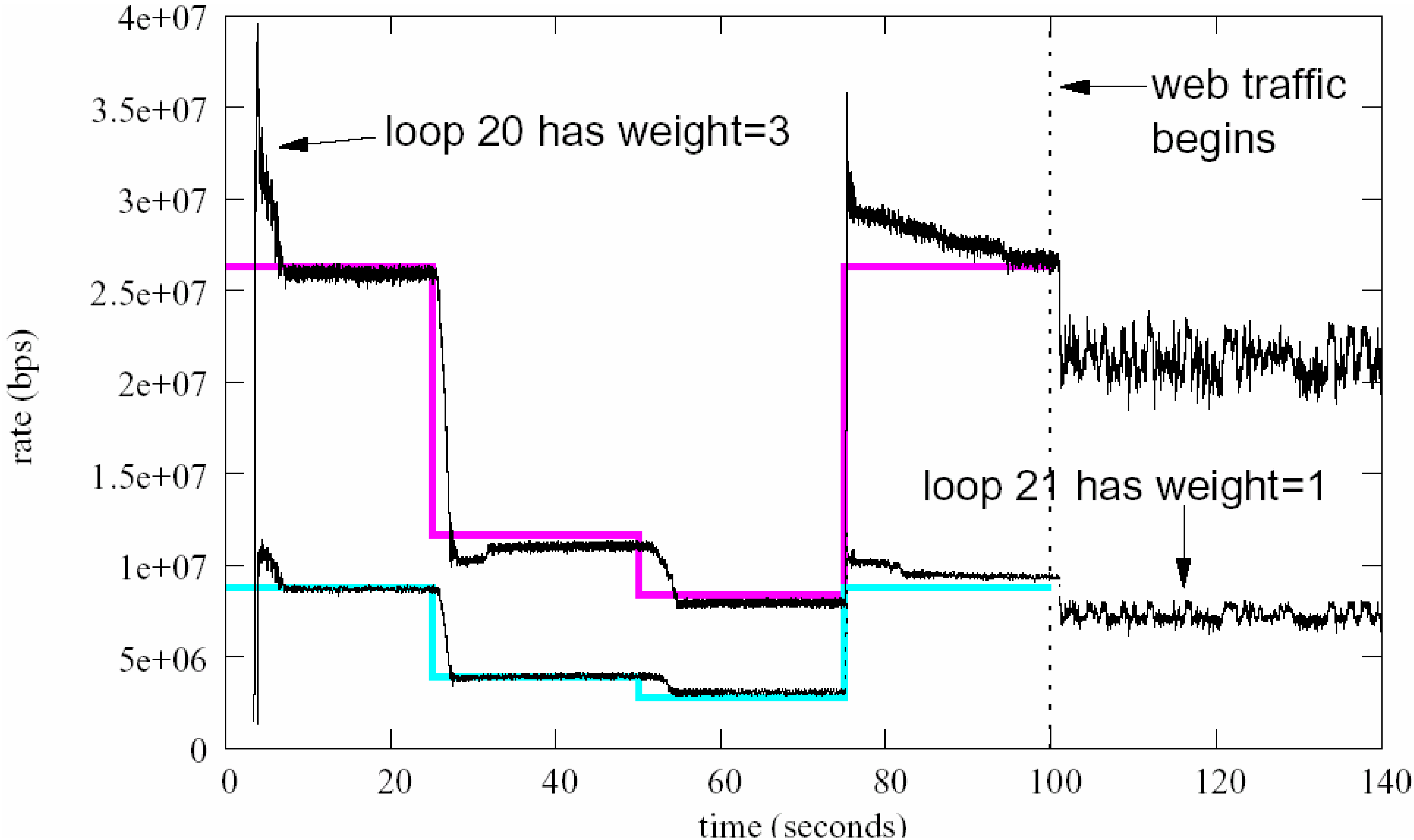
QoS spectrum

EXTRA SLIDES

Mean Queue Length for EMR



Service Multiplexing: Weighted shares



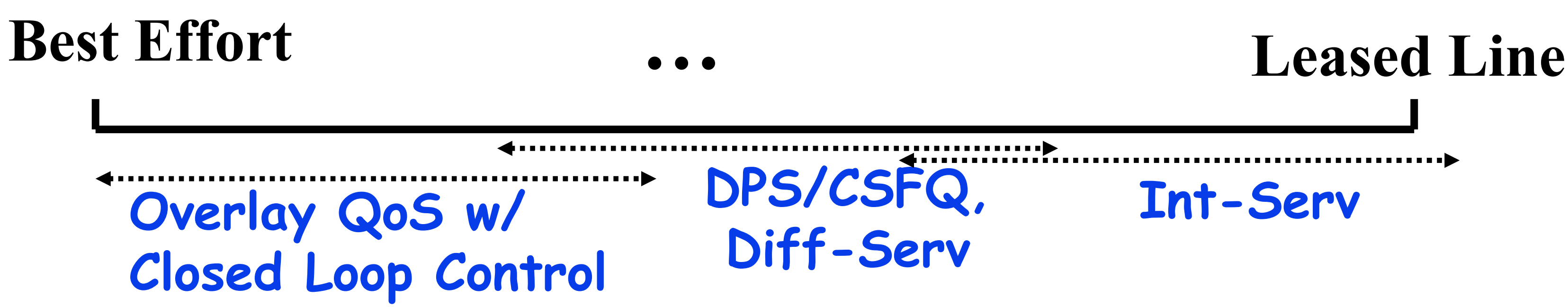
Rensselaer Polytechnic Institute
Shivkumar Kalyanaraman

In General: Closed-Loop => Better-than-Best-Effort Services

- A weaker/broader view of QoS:
- QoS: "**Better performance (given fixed routes)**":
  - Described ***a priori*** by a set of parameters **AND/OR**
  - Measured ***a posteriori*** by a set of metrics.

(extra slides on results if you are interested)

QoS spectrum



Summary: Closed-Loop QoS

- QoS can be viewed as a congestion control problem and posed in Kelly's optimization framework
  - Allows distributed admission control, or even services without admission control (distributed parameter choices).
  - Tradeoff: objectives achieved only in steady state
- Accumulation-based schemes (eg: Monaco) provide a physically meaningful steering parameter (accumulation) relating to queue length
  - Which is also the lagrange multiplier, and
  - Is the weight parameter in weighted proportional fairness allocation
  - Requires an extra priority queue for control pkts (IP precedence)
  - AQM support => virtual accumulation => ~0 queues
- Convex constraints on accumulation, queue length (I.e. in lagrange multiplier domain):
  - assures unique optimum; and
  - leads to graceful degradation of service assurances
- Schemes implemented on Linux and tested in Utah Emulab – to be deployed in PlanetLab
- Developing multimedia applications to leverage these lightweight QoS capabilities along with multi-path capabilities in an overlay network

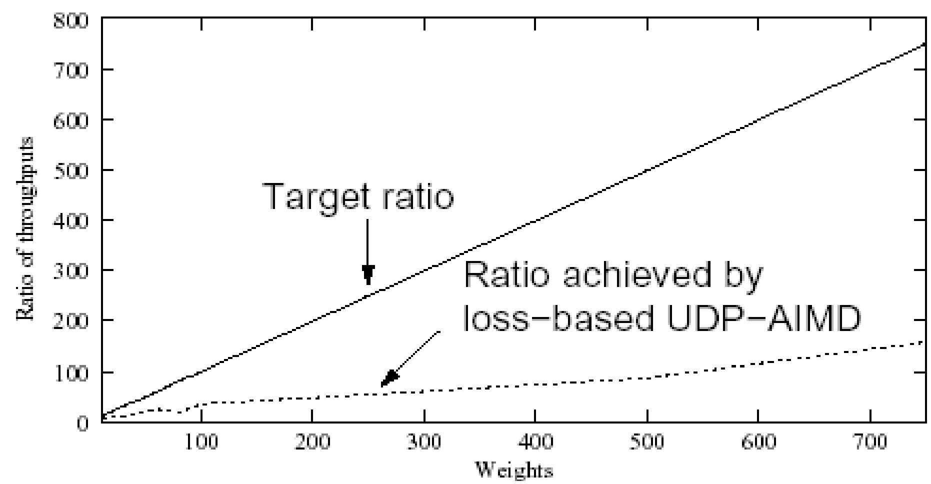
EMR Algorithm (for reference)

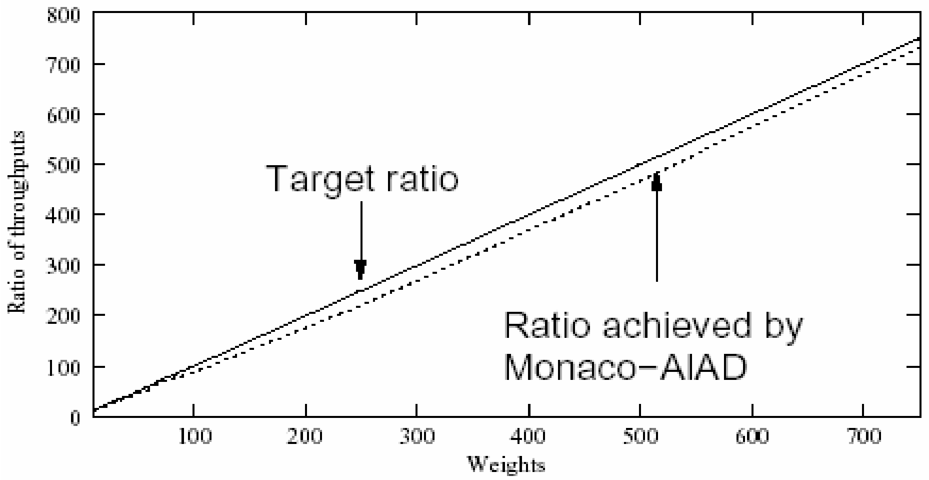
Algorithm 1 Expected Service Pseudo-code at Ingress

$cwnd$ = the congestion window in bytes
$pwnd$ = the congestion window in the previous RTT
$ssthresh$ = the slow start threshold
$srtt$ = the smoothed RTT estimation
A = the total accumulation limit
ε = the target accumulation beyond the expected minimum rate

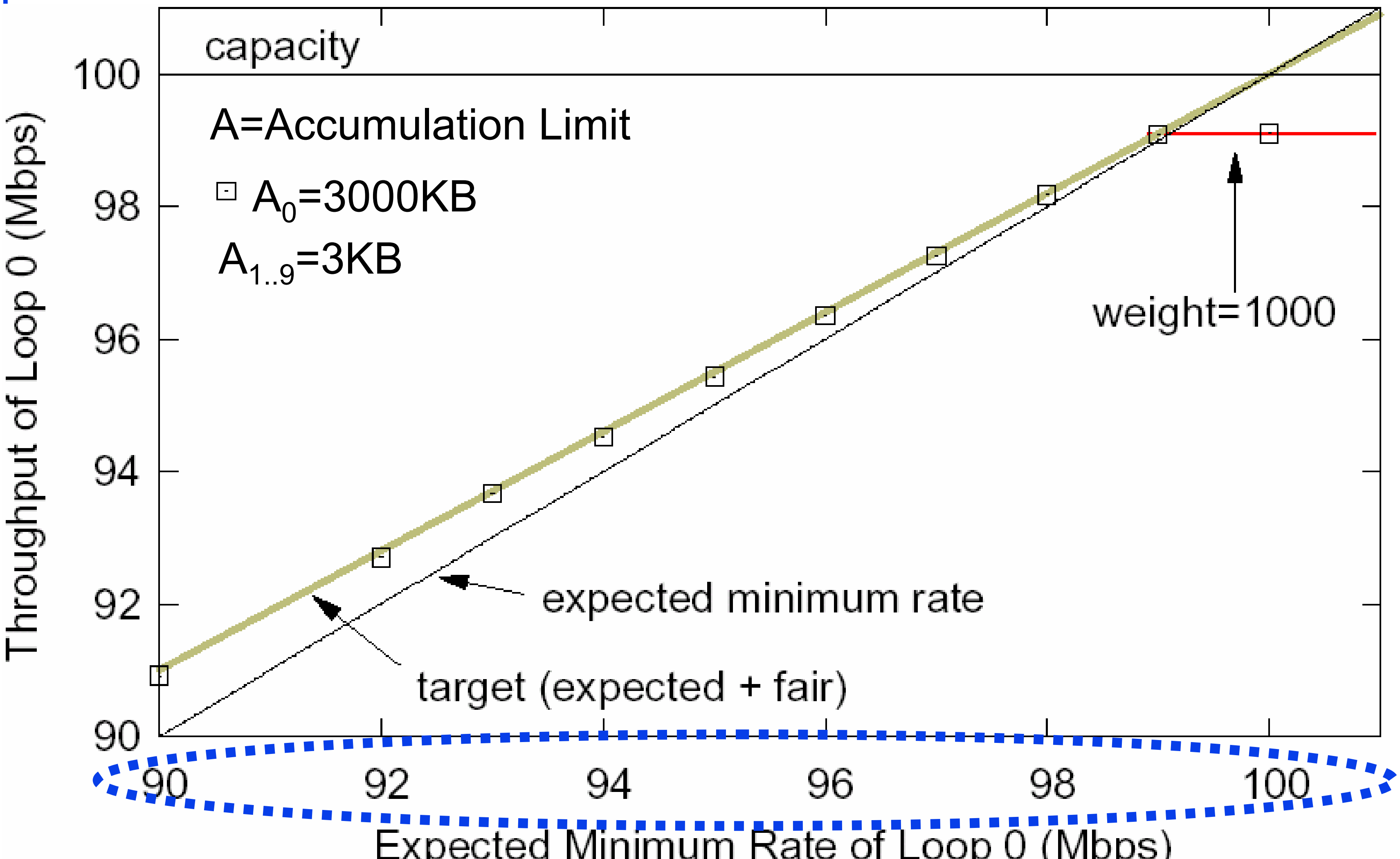
(1) a = reverse_ctrl_pkt.accumulation;
(2) $x = pwnd * 8.0/srtt;$
(3) $a_p = max(a * (1 - x_e/x), 0.0);$
(4) $pwnd = min(pwnd + mtu, cwnd);$
(5) $cwnd = pwnd - k * max(a_p - \varepsilon, a - A);$
(6) if $(a > A \,||\, a_p > \varepsilon)$ $\{ ssthresh = cwnd; \}$
(7) else {
    (7.1) if $(pwnd + mtu >= ssthresh)$
        $ssthresh = cwnd;$
    (7.2) $cwnd = min(pwnd * 2.0, ssthresh);$ }
(8) rate_limit = $cwnd * 8.0/srtt;$

Eg: Weighted Service w/ Loss-based vs Accumulation-based schemes

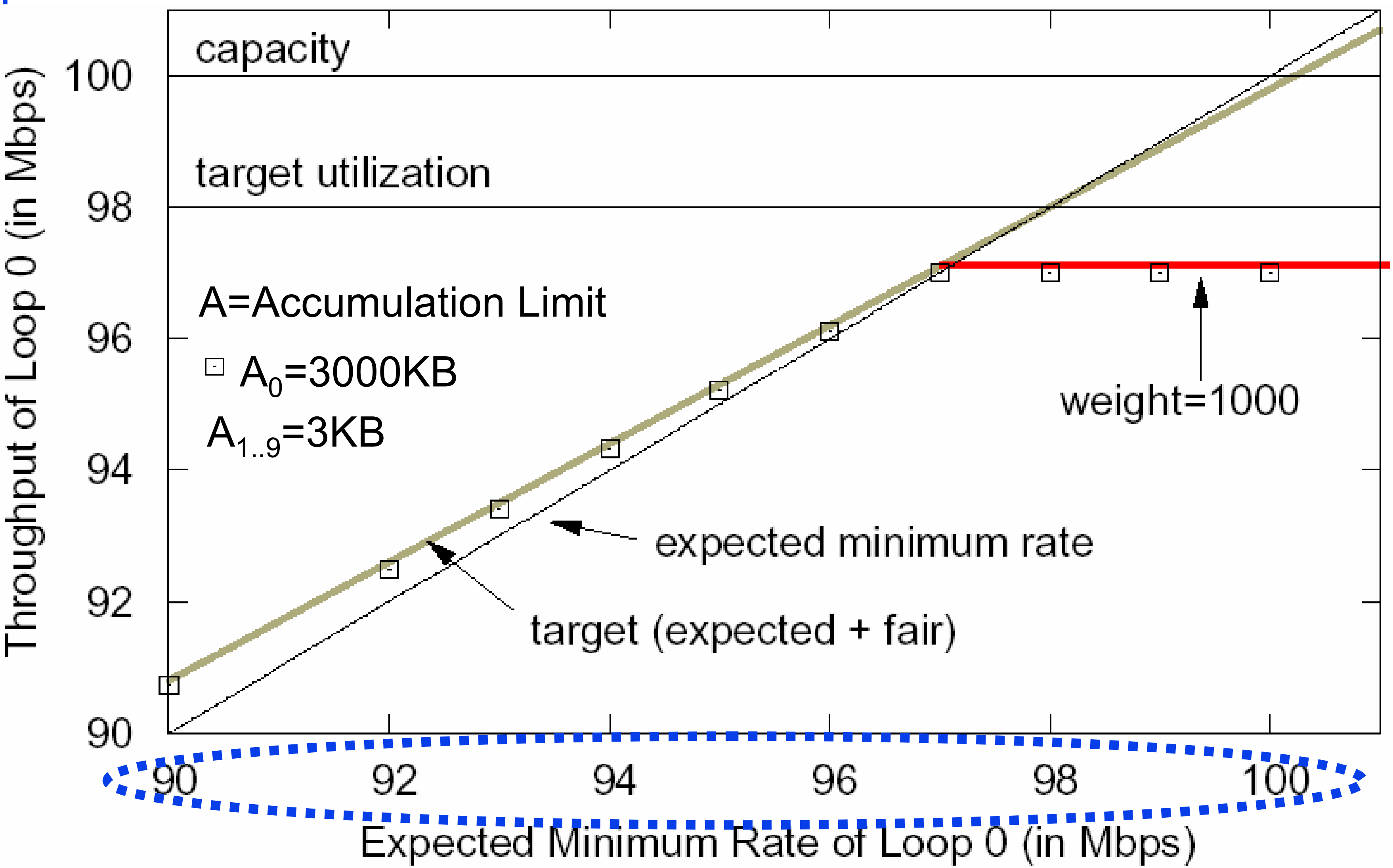




No AQM and EMR Near Full Capacity

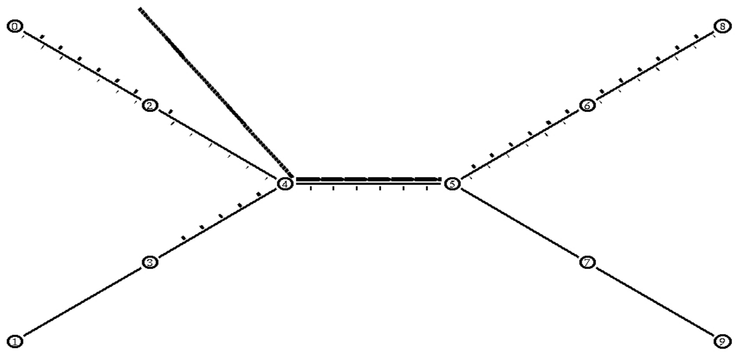


AVQ+VD+ EMR Near Full Capacity

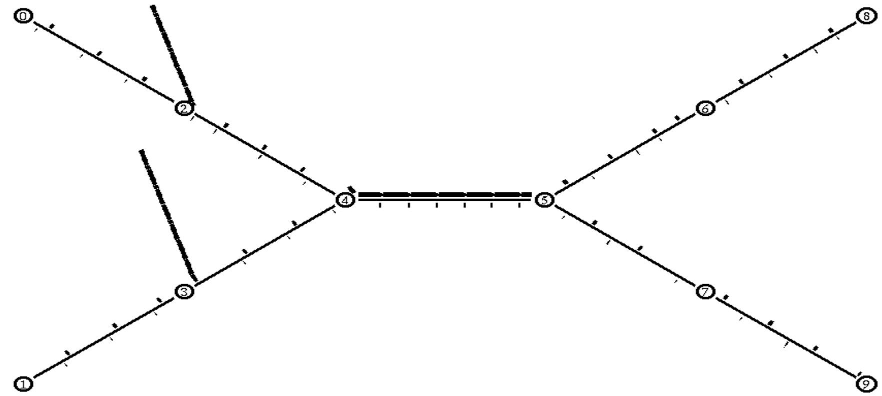


Scalable Best-effort TCP Service

Without Overlay Scheme



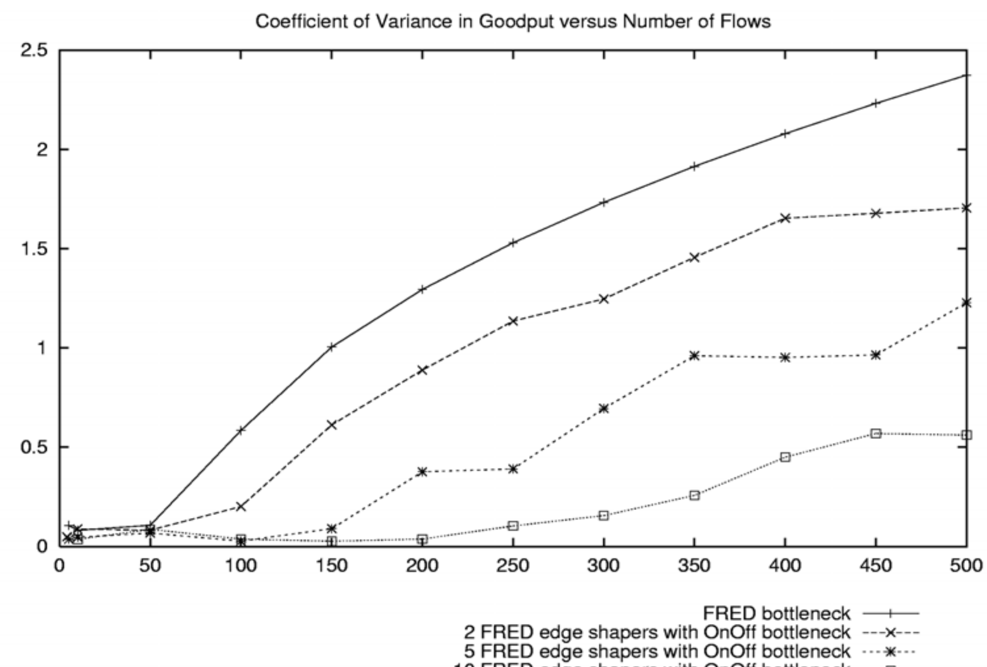
With Overlay Scheme



Queue distribution to the edges => can manage more efficiently

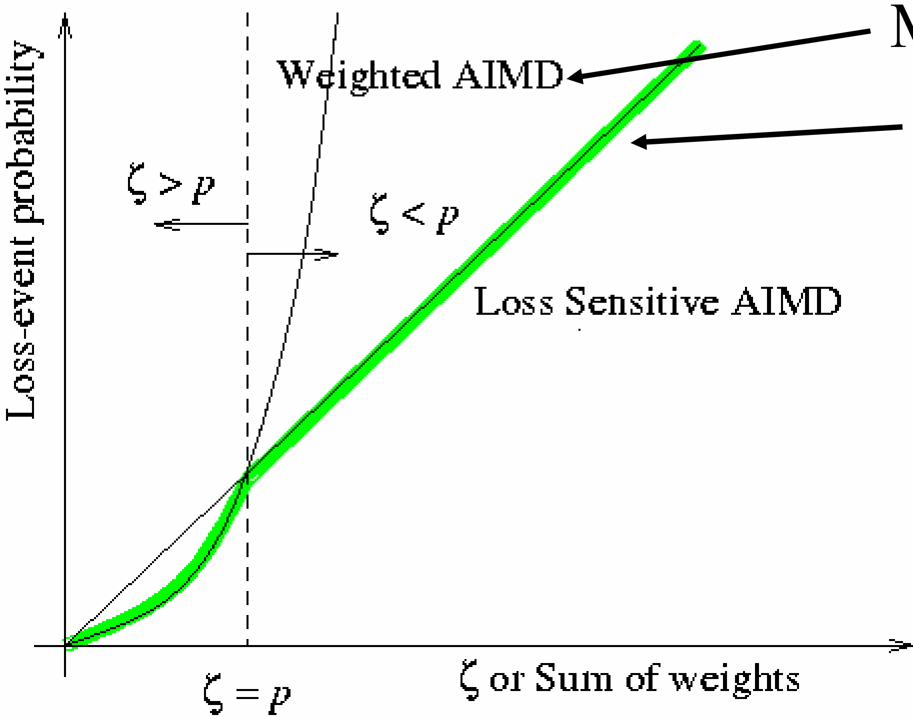
CoV vs. No of Flows

FRED at the core ***vs***. FRED at the edges with overlay control between edges



Weighted Sharing

- Proposed many times (MulTCP, TCP-SD, TCP-LASD, IP-Trunking, Nonlinear Optimization-based Congestion Control).
- MulTCP and TCP-SD use loss-based differentiation.

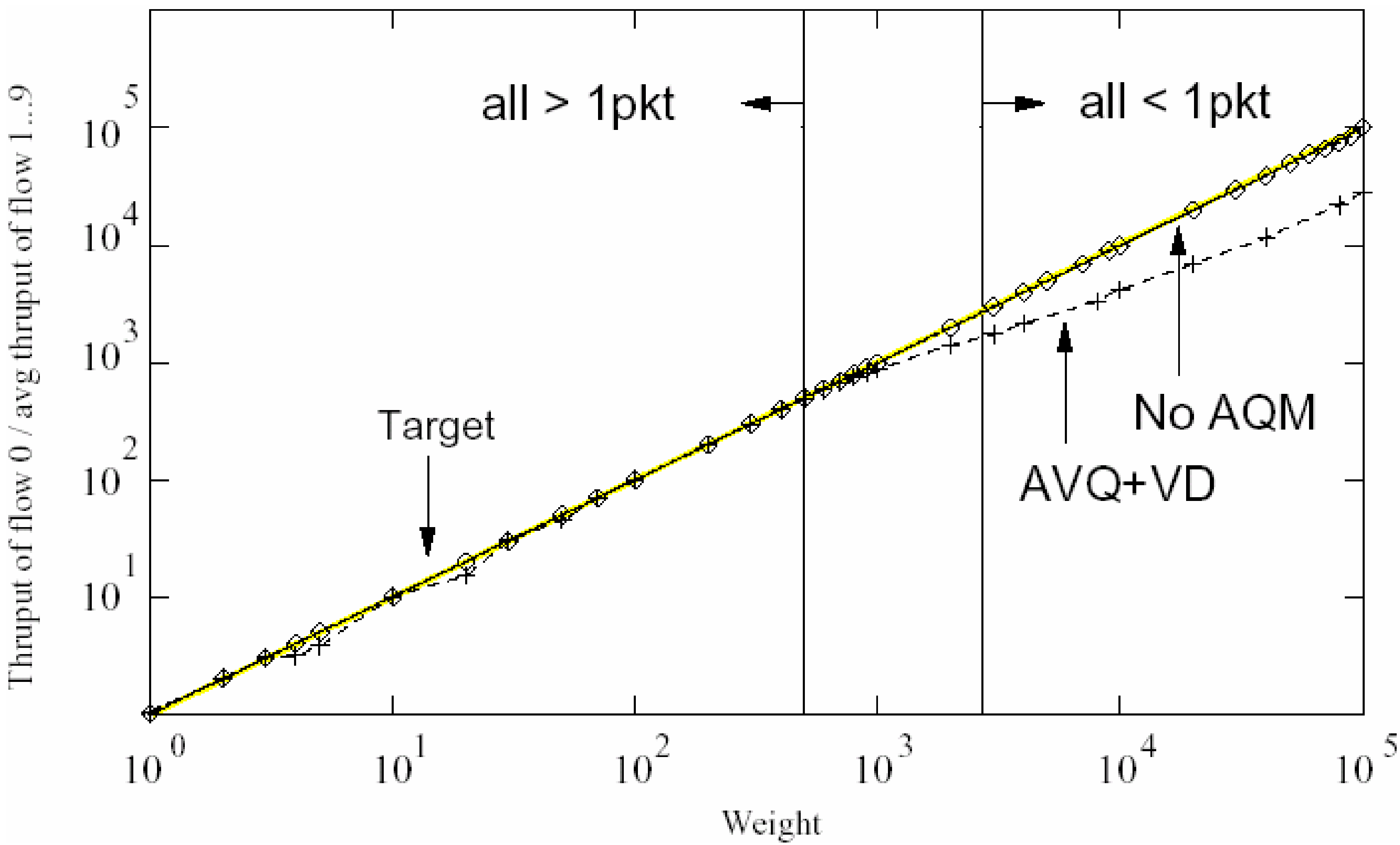


MulTCP,TCP-SD loss $\propto \left(\sum W_i\right)^2$

TCP-LASD loss $\propto \sum W_i$

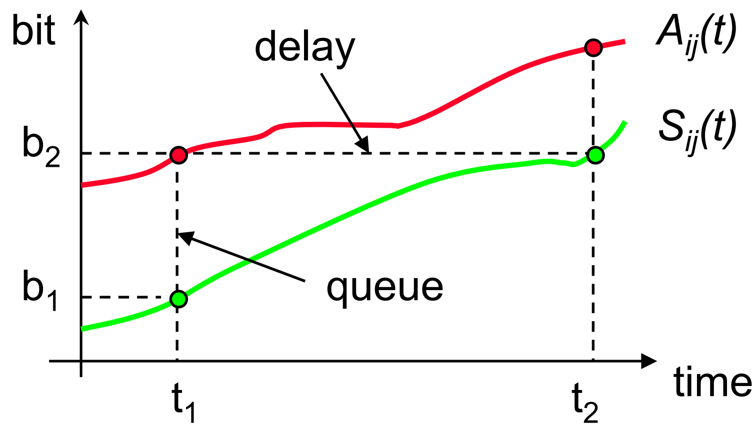
- Heavy weights → more aggressive in response to loss → heavy loss
- Timeouts limit range of weights achievable (10-to-1 MulTCP, 100-to-1 TCP-LASD).

Range of Weighted Services



Δ(Flow's Queue Contribution) at One FIFO Router

- flow i at router j
- arrival curve $A_{ij}(t)$ & service curve $S_{ij}(t)$
  - cumulative
  - continuous
  - non-decreasing



- if no loss, then

$$\because q_{ij}(t) = A_{ij}(t) - S_{ij}(t)$$

$$\therefore q_{ij}(t+\Delta t) = A_{ij}(t+\Delta t) - S_{ij}(t+\Delta t)$$

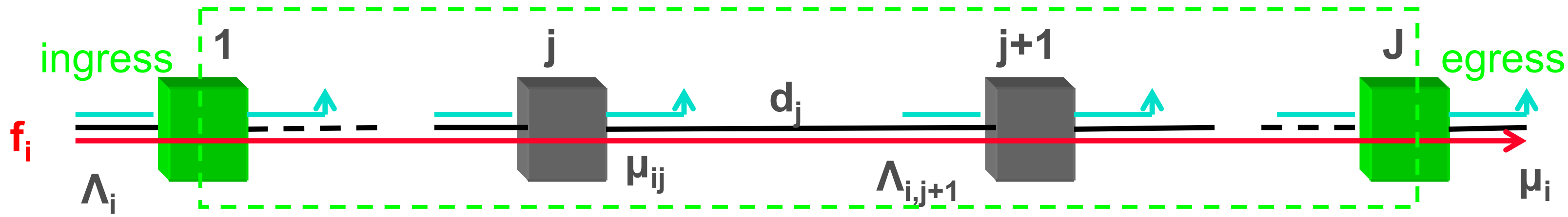
$$\therefore \Delta q_{ij}(t,\Delta t) = q_{ij}(t+\Delta t) - q_{ij}(t)$$

$$= [A_{ij}(t+\Delta t) - A_{ij}(t)] - [S_{ij}(t+\Delta t) - S_{ij}(t)]$$

$$= [\overline{\lambda}_{ij}(t,\Delta t) - \overline{\mu}_{ij}(t,\Delta t)] \times \Delta t$$

$$\boxed{= I_{ij}(t,\Delta t) - O_{ij}(t,\Delta t)}$$

Δ(Accumulation): Series of FIFO Routers



- then $\Delta a_i(t, \Delta t) = a_i(t + \Delta t) - a_i(t)$

$$= \sum_{j=1}^{J} q_{ij}(t + \Delta t - \sum_{k=j}^{J-1} d_k) - \sum_{j=1}^{J} q_{ij}(t - \sum_{k=j}^{J-1} d_k)$$

$$= \sum_{j=1}^{J} \Delta q_{ij}(t - \sum_{k=j}^{J-1} d_k, \Delta t)$$

$$= \sum_{j=1}^{J} [\overline{\lambda_{ij}}(t - \sum_{k=j}^{J-1} d_k, \Delta t) - \overline{\mu_{ij}}(t - \sum_{k=j}^{J-1} d_k, \Delta t)] \times \Delta t$$

$$= [\overline{\lambda_i}(t - d_i^f, \Delta t) - \overline{\mu_i}(t, \Delta t)] \times \Delta t$$

$$\boxed{= I_i(t - d_i^f, \Delta t) - O_i(t, \Delta t)}$$

where $d_i^f = \sum_{j=1}^{J-1} d_j$